NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

AUTONOMOUS AGENT-BASED SIMULATION OF A MODEL SIMULATING THE HUMAN AIR-THREAT ASSESSMENT PROCESS

by

Baris Egemen OZKAN

March 2004

Thesis Co-Advisors:	John Hiles
	Neil Rowe
Second Reader:	Chris Darken

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** March 2004	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE: Autonomous Agent-Based Simulation Of A Model Simulating The Human Air-threat Assessment Process	**5. FUNDING NUMBERS**
6. AUTHOR(S) Baris Egemen OZKAN	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

The Air Defense Laboratory (ADL) Simulation is a software program that models the way an air-defense officer thinks in the threat assessment process. The model uses multi-agent system (MAS) technology and is implemented in Java programming language. This research is a portion of Red Intent Project whose goal is to ultimately implement a model to predict the intent of any given track in the environment. For any air track in the simulation, two sets of agents are created, one for controlling track actions and one for predicting its identity and intent based on information received from track, the geopolitical situation and intelligence. The simulation is also capable of identifying coordinated actions between air tracks. We used three kinds of aircraft behavior in the simulation: civilian, friendly and enemy. Predictor agents are constructed in a layered structure and use "conceptual blending" in their decision-making processes using mental spaces and integration networks. Mental spaces are connected to each other via connectors and connecters trigger tickets. Connectors and Tickets were implemented using the Connector-based Multi Agent System (CMAS) library. This thesis showed that the advances of Cognitive Science and Linguistics can be used to make our software more cognitive. This simulation is one of the first applications to use cognitive blending theory for a military application. We demonstrated that agents can create an "integration network" composed of "mental spaces" and retrieve any mental space data inside the network immediately without traversing the entire network by using the CMAS library. The results of the tests of the simulation showed that the ADL Simulation can be used as assistant to human air-defense personnel to increase accuracy and decrease reaction time in naval air threat assessment.

14. SUBJECT TERMS Air Defense Laboratory Simulation, Naval Air-threat Assessment, Air-defense, Conceptual Blending Theory, Air-defense Simulation, Multi Agent Systems	**15. NUMBER OF PAGES** 120
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

THIS PAGE INTENTIONALLY LEFT BLANK

AUTONOMOUS AGENT-BASED SIMULATION OF A MODEL SIMULATING THE HUMAN AIR-THREAT ASSESSMENT PROCESS

Baris E. OZKAN
Lieutenant Junior Grade, Turkish Navy
B.S., Turkish Naval Academy, 1998

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
March 2004

Author:	Baris Egemen OZKAN
Approved by:	John Hiles Thesis Co-Advisor
	Neil Rowe Thesis Co-Advisor
	Chris Darken Second Reader
	Peter J. Denning Chairman, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The Air Defense Laboratory (ADL) Simulation is a software program that models the way an air-defense officer thinks in the threat assessment process. The model uses multi-agent system (MAS) technology and is implemented in Java programming language. This research is a portion of Red Intent Project whose goal is to ultimately implement a model to predict the intent of any given track in the environment. For any air track in the simulation, two sets of agents are created, one for controlling track actions and one for predicting its identity and intent based on information received from track, the geopolitical situation and intelligence. The simulation is also capable of identifying coordinated actions between air tracks. We used three kinds of aircraft behavior in the simulation: civilian, friendly and enemy. Predictor agents are constructed in a layered structure and use "conceptual blending" in their decision-making processes using mental spaces and integration networks. Mental spaces are connected to each other via connectors and connecters trigger tickets. Connectors and Tickets were implemented using the Connector-based Multi Agent System (CMAS) library. This simulation is one of the first applications to use cognitive blending theory for a military application. We demonstrated that agents can create an “integration network” composed of “mental spaces” and retrieve any mental space data inside the network immediately without traversing the entire network by using the CMAS library. The results of the tests of the simulation showed that the ADL Simulation can be used as assistant to human air-defense personnel to increase accuracy and decrease reaction time in naval air-threat assessment.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION 1
- A. THE AIR-DEFENSE LABORATORY SIMULATION 1
- B. MOTIVATION FOR THE ADL SIMULATION 2
- C. MULTI-AGENT SYSTEMS IN ADL SIMULATION 4

II. CONCEPTUAL BLENDING THEORY 9
- A. INTRODUCTION TO CONCEPTUAL BLENDING THEORY 9
- B. FORMS 9
- C. PRINCIPLES OF BLENDING 10
- D. NETWORKS OF SPACES IN BLENDING THEORY 13
  - 1. Simplex Networks 16
  - 2. Mirror Networks 17
- E. NETWORKS IN THE ADL SIMULATION 18
- F. ALTERNATIVE WAYS TO BLENDING THEORY AND CMAS LIBRARY FOR ADL SIMULATION 19

III. RELATED WORK IN NAVAL AIR-THREAT ASSESSMENT 21
- A. RELATED WORK INTRODUCTION 21
- B. ADVERSARIAL PLAN RECOGNITION FOR AIRBORNE THREATS 21
- C. NAVAL AIR-DEFENSE THREAT ASSESSMENT: COGNITIVE FACTORS AND MODEL 22
- D. AIR-THREAT ASSESSMENT: RESEARCH, MODEL AND DISPLAY GUIDELINES 24
- E. SIMULATION OF AN AEGIS CRUISER COMBAT INFORMATION CENTER 25
- F. MULTISENSOR DATA FUSION 27
- G. MULTISENSOR DATA FUSION 28
- H. COMPARISON WITH OTHER AIR-DEFENSE WORK 29

IV. MULTI-AGENT SYSTEMS AND DESCRIPTION OF AIR DEFENSE LABORATORY SIMULATION 31
- A. PROGRAM LANGUAGE AND SYSTEM REQUIREMENTS FOR ADL SIMULATION 31
- B. MULTI-AGENT SYSTEMS AND THE CMAS LIBRARY 31
  - 1. Agents 31
  - 2. Connector Based Multi-Agent Systems and CMAS Library 32
    - *a. Connectors 32*
    - *b. Tickets 34*
    - *c. Ticket & Connector Structures 35*
    - *d. The CMAS Library 36*
- C. MENU OPTIONS 36

D. TOOLBAR OPTIONS.......... 42
E. OUTPUT PANEL.......... 44
F. JAVA XML INTEGRATION.......... 44

V. DESIGN OF THE ADL SIMULATION PROGRAM.......... 47
A. INTRODUCTION.......... 47
B. REAL TRACK AGENTS.......... 47
1. Civilian Aircraft Track Agent.......... 48
2. Friendly Aircraft Track Agent.......... 49
3. Enemy Aircraft Track Agent.......... 50
4. Coordinated Detachment Attack Track Agent.......... 52
5. Missile Attack Track Agent.......... 53
6. Missile Track Agent.......... 54
7. User derived Track Agent.......... 54
C. REACTIVE AGENTS.......... 54
1. Airlane Reactive Agent.......... 55
2. ESM Reactive Agent.......... 56
3. Heading-change Reactive Agent.......... 57
4. IFF Reactive Agents.......... 57
5. Max Acceleration, Altitude, and Speed Reactive Agents... 58
6. Origin Reactive Agent.......... 60
7. Radar Status Reactive Agent.......... 60
8 Random Number Finder Reactive Agent.......... 60
9. Snooper Detector Reactive Agent.......... 62
D. PREDICTOR TRACK AGENTS.......... 63
1. Predictor Agent Connectors and Queries.......... 64
2. Predictor Agent Competing Models.......... 66
3. Predictor Agent Tickets.......... 66
4. Weighting Procedure for the Predictor Track Agent.......... 71
a. Weighting the Civilian Ticket.......... 72
b. Weighting the Friendly Ticket.......... 72
c. Weighting the Hostile Ticket.......... 73
d. Weighting the Suspect Ticket.......... 75
e. Weighting the Unknown Ticket.......... 75
f. Execution of the ATO Ticket.......... 75
g. Execution of IFF Evaluation Ticket.......... 76
h. Execution of CPA Ticket.......... 77
i. Split Activity Detection.......... 77
E. THE REGIONAL AGENT.......... 78
F. BLENDING THEORY AND THE ADL SIMULATION.......... 81

VI. RESEARCH QUESTIONS RESULTS AND THE EVAULATION OF THE SIMULATION.......... 85
A. RESEARCH QUESTIONS.......... 85
1. Overview.......... 85
1. General Testing Methodology.......... 85
B. THE LEVEL OF REALITY OF THE ADL SIMULATION.......... 86

C. THE ACCURACY OF THE DECISIONS BY THE ADL SIMULATION 87
D. THE LEVEL OF CLOSENESS OF DECISIONS GIVEN BY THE MODEL TO THE DECISIONS GIVEN BY THE EXPERTS 90

VII. FUTURE WORK AND DEVELOPMENT OF THE AIR DEFENSE LABORATORY SIMULATOR 91
A. FUTURE WORK INTRODUCTION 91
B. DEVELOPMENT OF THE ENVIRONMENT 91
C. INTEGRATED KNOWLEDGE TRANSFER 91
D. THE ADL SIMULATION AS A TRAINING TOOL 92
E. IMPLEMENTING THE ADL SIMULATION WITH BAYESIAN METHODS 93

VIII. SUMMARY AND CONCLUSION 95

LIST OF REFERENCES 97

INITIAL DISTRIBUTION LIST 101

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Figure 1. ADL Simulation Interface 2
Figure 2. ADL Simulation MAS Layout 5
Figure 3. Conceptual Blending (After [1]) 11
Figure 4. Conceptual Integration Network 14
Figure 5. Simplex Network (After [1]) 16
Figure 6. Mirror Network (After [1]) 18
Figure 7. Cognitively-Based Model of Threat Assessment (From [20]) 23
Figure 8. Threat Assessment Model (From [22]) 24
Figure 9. Contact Classification Artificial Neuron (From [24]) 26
Figure 10. Connector States 33
Figure 11. Visual Design of the ADL Simulation 37
Figure 12. Tactical Figure Control Panel 38
Figure 13. Track Generator Panel 38
Figure 14. Coordinated Detachment Attack 39
Figure 15. Evaluation of Model Panel 41
Figure 16. Track Info pages 42
Figure 17. Regional Agent and Track Agent Connectors Frames 43
Figure 18. Competing Models Frame 43
Figure 19. Simulation Output Panel 44
Figure 20. Coordinated Detachment Attack Profile 53
Figure 21. ADL Simulation Layered Structure 55
Figure 22. ADL Random Number Finder Reactive Agent Equation 61
Figure 23. Snooper Detector Reactive Agent Equation 63
Figure 24. Civilian Ticket and Frames 67
Figure 25. A Friendly Ticket and Frames 68
Figure 26. Hostile Ticket and Frames 68
Figure 27. Suspect Ticket and Frames 69
Figure 28. Unknown Ticket and Frames 69
Figure 29. Connecting Local Independent Ticket Integration Network to Identity Integration Network 70
Figure 30. IFF Evaluation Independent Ticket 71
Figure 31. Evaluation of ATO Frame of Hostile Ticket 76
Figure 32. Predictor Agent Split Activity Detector Ticket 77
Figure 33. Regional Agent Snooper Detector Ticket 80
Figure 34. Merge Detector Blending Operation 81

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1. Default Classification Threshold Values(From [24])............................ 26
Table 2. Scoring (Weighted) Values for Various Input Cues(From [24])........... 27
Table 3. Civilian Aircraft Attributes and Behaviors.. 48
Table 4. Friendly Aircraft Missions [28] 49
Table 5. Friendly Aircraft Attributes and Behaviors 50
Table 6. Enemy Aircraft Missions [28].. 51
Table 7. Enemy Aircraft Attributes and Behaviors.. 52
Table 8. ESM Reactive-agent Messages to Track Agent................................. 57
Table 9. IFF System Modes and Concepts 58
Table 10. Sample Error Percentage Limits... 61
Table 11. Predictor Track Agent Connectors and Queries................................. 66
Table 12. Regional Agent Connectors and Queries 79
Table 13. Scenarios Used in the Simulation Test and Analysis.......................... 86
Table 14. Civilian Aircraft ID Results of Tests 88
Table 15. Friendly Aircraft ID Results of Tests 88
Table 16. Hostile Aircraft ID Results of Tests... 89

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

This thesis is the culmination of my military experience and education. As an Anti-Air Warfare Officer (AAWO) of TCG ORUCREIS (F-245) and a master degree student at Naval Postgraduate School, I was able to blend both theory and application from my personal experience to my thesis. I have combined my naval air defensive warfare experience with the computer science knowledge that I gained at Naval Postgraduate School. I am grateful to everyone who helped me in achieving my goals especially my senior officers who taught me air warfare concepts, my air-defense team on the ship, and all of my instructors at the Naval Postgraduate School. Special thanks to Prof. John Hiles, Prof. Neil C. Rowe and Prof. Chris Darken.

I also would like to thank my family members for their endlessly supporting me and for being in my life. I am so grateful to my mom for showing me how to be strong in life and keep up with the struggles of life by beating two different cancers. I am very thankful to my father for inspiring me to one day be as a great father as he is. I am also thankful to Basar, my brother, for being the best brother imaginable.

I also would like to thank my girlfriend, Jennifer M. Courtney, for joining my life and sharing her life with me. Her help and support was key in the completion of my thesis.

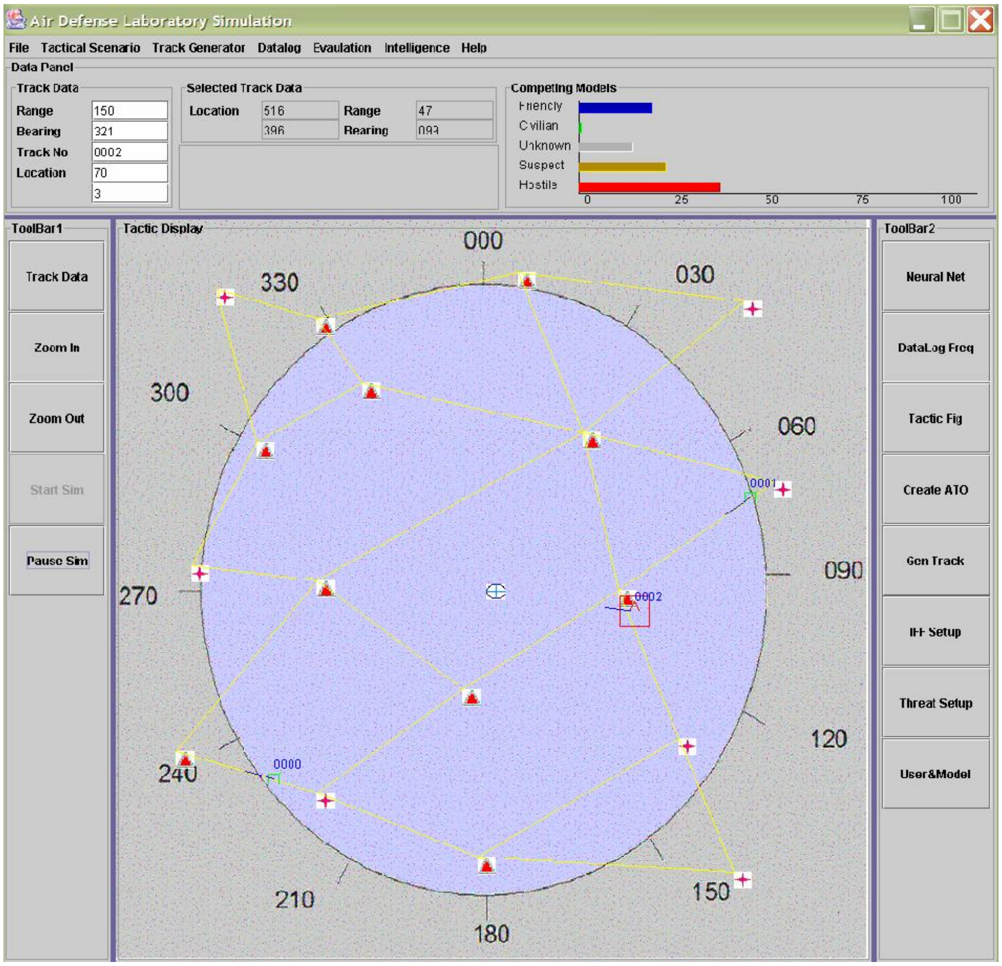
THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. THE AIR-DEFENSE LABORATORY SIMULATION

The Air Defense Laboratory (ADL) Simulation is a software program which simulates an Anti-Air Warfare (AAW) Officer's threat assessment process in the Combat Information Center (CIC) of a frigate performing air defense. ADL is a software cognitive model implemented in the Java programming language. It is user-interactive in that it allows users to manipulate input data and create realistic air-defense scenarios. The program simulates the mental processes performed by an AAW Officer in the threat-assessment phase. It uses multiagent systems technology and the Connector Based Multi-Agent Systems (CMAS) Library written by the Integrated Asymmetric Goal Organization (IAGO) team at the Naval Postgraduate School. The cognitive model implements Conceptual Blending Theory as proposed by Turner and Fauconnier[1].

The ADL Simulation has two goals, one short-term and one long-term. In the short term, the simulation aims to assist air-defense teams to gain insight about air-threat assessment and to support the team in decision-making under stressful conditions. The air-defense crew may create queries for a specific air track and receive advice and predictions about the possible intentions of the track from the simulation. Differently from most other approaches in which the only final decision is presented, the ADL Simulation also gives the user reasons as to why the steps towards a decision are made by traversing backwards each node of an "integration network" that is created in the model. In the long term, the ADL Simulation aims to improve decisionmaking and air-threat assessment duties of the Anti-Air Warfare Officer. Currently in naval technology, the primary use of unmanned sensory vehicles is reconnaissance. In the future, when the cognitive models like that of the ADL Simulation are embedded in these vehicles, they will be able to make decisions and take actions in the field. This would save much time, money, and human resources that are currently used. This would also limit the placement of humans in dangerous situations and the loss of life or additional risks that accompany these decisions.



Air Defense Laboratory Simulation

Data Panel

Track Data

Range: 150
Bearing: 321
Track No: 0002
Location: 70
3

Selected Track Data

Location: 516
396
Range: 47
Bearing: 093

Figure 1. ADL Simulation Interface

B. MOTIVATION FOR THE ADL SIMULATION

Naval air warfare is the most rapid and intense traditional warfare. If not properly executed it could result in severe destruction. The attacker has the advantage of speed, flexibility of the attack axis, direction, and time. With a well-coordinated attack, a ship's self-defense systems may also be saturated with overwhelming data. This requires that a naval unit focus on expertly training their air-defense teams and maximize their proficiency level.

Currently there are two types of training for air-defense teams. The first is the training with actual aircraft at sea and the second is training with simulated tracks. Even though the simulators are well designed and represent the reality of air warfare, they suffer from being a simulation in the air-defense team’s minds and do not provide the realistic atmosphere and the factors associated stressful conditions. Realistic training requires the use of the actual aircraft and other relevant training components, a huge amount of resources, and pre-coordination between the air assets and the naval unit. A naval force without an aircraft carrier has to arrange all of this coordination. Most of the time, training with real air assets is short due to the limited flight time of aircraft. This motivates the need for new simulators better representing the real world to assist air-defense teams in naval air warfare.

Another way to increase success in defensive air warfare and to compensate for the disadvantage of being on the defensive side is to assist the air-defense team in the command-and-control systems in the CIC. These embedded systems support the air-defense crew in threat assessment by showing threat priorities, sorting threats based on priorities, and reminding the air-defense team of the Anti-Ship Missile Defense (ASMD) procedures. There has been a great deal of work done to increase the efficiency of the supporting software embedded in the command-and-control systems. This is especially relevant because of two high-profile incidents concerning the USS Vincennes and USS Stark. On March 17, 1987 a Mirage F-1 fighter jet, took off from Iraq's Shaibah military airbase. It was detected by both USS Stark and an Airborne Warning and Control System (AWACS) plane. The attacker Mirage released its load and headed to the north but the USS Stark and the AWACS did not detect the missiles. On July 3, 1988, 15 months after the previous incident, USS Vincennes shot down a civilian Iranian Airliner carrying 290 people after falsely identifying it as an attacking aircraft. The reports released after these two events clearly revealed the importance of the human factor in air defense. In both incidents, the lack of correct decision-making about the situation and situational awareness had a catastrophic result.

After these incidents, U.S. Navy research focused on assisting humans in air defense using the fields of artificial intelligence (AI) and display technologies. The ADL Simulation deals mostly with the AI and aims to assist the air-defense crew. The post-Stark and post-Vincennes research supporting air-defense teams in the AI field is discussed in Chapter III.

C. MULTI-AGENT SYSTEMS IN ADL SIMULATION

The ADL Simulation uses Multi-Agent System (MAS) technology. “A basic Multi Agent System (MAS) is an electronic or computing model made of artificial entities that communicate with each other and act in an environment.” [2] Agents are autonomous software elements operating and interacting with each other in that environment.

MAS’s are comprised of six components: an environment (E) which is the space where agents operate, the objects (O) in the environment, actors (A), relations (R) between actors and the environment and between agents, operations (Ops) that are executed by actors, and laws of the environment.

MAS = {E, O, A, R, Ops, Laws}

Agents in a MAS environment receive input from the environment, process this input and produce an output. This output is eventually released to the environment by the agents. The environment may have more than one agent in it and provides communication facilities to all agents. Its architecture is usually formulated as sense-process-act.

The environment in the ADL Simulation is the airspace containing air assets and the air-defense ship equipped with satisfactory air-defense sensors. One class of objects are the sensory devices of the ship which provide input information to threat assessors. There are mainly two kinds of agents in ADL Simulation: Real-track agents (offensive) and predictor agents (defensive). Real-track agents control aircraft activities based on the type of the aircraft. Predictor agents receive sensory data produced by real-track agents and generate a prediction about the identity and possible intent of the aircraft. Predictor agents are designed in a layered architecture. At the very bottom level of prediction, reactive agents reside. They are responsible for each factor in air-threat assessment, discussed in detail in Chapters III and V. The track agents are located above reactive agents and are responsible for combining all information provided by reactive agents. Regional agents are above the track agents and are responsible for identifying coordinated activities between air tracks. Communication between agents is provided by connectors implemented with the CMAS library, as discussed in detail in Chapter V.

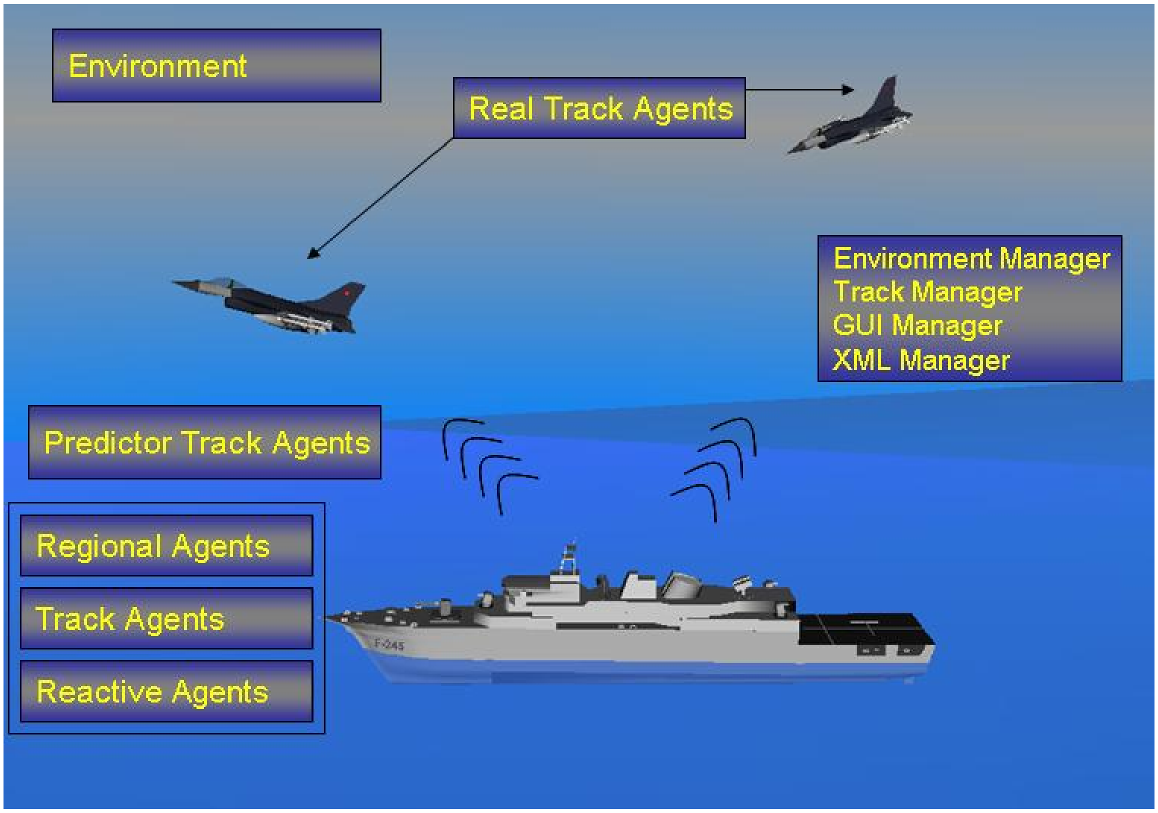


Figure 2. ADL Simulation MAS Layout

Predictor agents figure out the identity of the aircraft and their possible intentions. There are four kinds of aircraft in the simulation: civilian, military friendly, military hostile and user-defined ones. Civilian and friendly aircraft take no hostile action and are generated randomly by the Track Manager. User-defined aircraft are generated via the user interface.

The time difference between the initial detection of an air track and the time when the air track represents a threat for the ship is relatively short in air warfare with respect to other kinds of warfare. This forces the air-defense team to evaluate all the sensory data including kinematics of aircraft, history data, the geopolitical situation, and intelligence in a short time and carefully. The team then must synthesize this information, make a decision about the identity of the aircraft, and take appropriate action. Actions are limited to the Rules of Engagements (ROE). But ROEs are strict guidelines and usually the actions are based on the identities of the contact of interests. Therefore the main task is to identify the air tracks and then look up the ROEs to take proper action. Wrong identification causes wrong actions as with the USS Vincennes incident. Air warfare leaves limited time to make the right decision. Today’s technology has brought us computers with ever-increasing speed that can help meet the time constraints of air warfare. They are also indifferent to stress which would eliminate factors related to human error, i.e. fatigue, making wrong decisions under stress, and lack of experience. Calfee in his master thesis at NPGS modeled the impact of fatigue, stress and experience on humans in air defense using software decisionmaking processes. The results evidently show the impacts of human related deficiencies on air warfare. For the above mentioned reasons, the ADL Simulation uses computers to resolve problems, help air-defense teams in identifying tracks, and making correct actions.

The ADL Simulation differs from previous research in that it uses conceptual blending theory for the cognitive model. This theory explains how the human brain constructs meaning in the mind. It was primarily developed in linguistics and all the examples provided by Turner and Fauconnier come from that area. The ADL Simulation is a software implementation of Blending Theory in a scientific field. The ADL Simulation has the advantage of using CMAS Library to simplify its modeling. Part of the library is support for Connectors and Tickets. Connectors are communication devices between agents in the environment and let us apply real-like world scenarios to software and create the integration network of Blending Theory. Connectors in CMAS library enable agents in the environment to communicate with each other without a direct relation or a global controller. Tickets are procedural instructions for agents and data organizing systems. Connectors and Tickets are discussed in Chapter IV. By using these techniques to anticipate the intent of an air contact, the simulation comes close to the ways that human CIC personnel create meaning that integrates the intent and the possible threat of the air contacts.

The results of the ADL Simulation showed that the agents in the simulation created an Integration network of which the nodes are mental spaces containing instant information. The agents made their decision as the way a human air-defense officer did in shorter time period and with same accuracy. The results opened a new way to extend this project to its second goal. Using CMAS library was key to developing the simulation and extending it to second goal.

THIS PAGE INTENTIONALLY LEFT BLANK

II. CONCEPTUAL BLENDING THEORY

A. INTRODUCTION TO CONCEPTUAL BLENDING THEORY

Conceptual Blending Theory proposed by Giles Fauconnier and Mark Turner is a theory of reasoning in the human brain [3]. How do we understand the things happening around us? How do we give meanings to the events? How do we combine multiple actions? For many years both scientists and non-scientific people have been searching for the answers. The short explanation to these questions involves evolution, an ongoing process that could last millions or billions of years. Most people use their reasoning ability to rationalize the events that happen around them without wondering how they are able to do this. We take advantage of the fact that we can think rather than questioning why we can. Conceptual blending theory is one way of explaining how we think, give meaning to what is happening around us, integrate this information, learn, and eventually gain experience with age. Blending is the key to this theory. Humans are constantly blending as they talk, imagine, listen and in every other action imaginable. It is integration of processes that we blend in our minds as we do all these activities.

B. FORMS

Forms are the most common way to represent things around us. One of the most commonly used forms is language. People communicate with each other with a complex system of forms known as languages. These languages may be either verbal languages like English, German, Spanish or some symbolic languages like Morse code, flags, searchlights, ASL (American Sign Language), or even smoke. We construct sentences, sentences are composed of words, words consist of letters, and letters are nothing but little points and lines drawn in a particular shape with an associated phoneme. What actually makes everybody come to the decision that “a” is “a” is not coming from the nature of the “letter a”; which is the combination of some points and lines. It is actually the “form” that we wrap around the “letter a”. Since everyone in the world who speaks a Latin language knows this form, “a” has the same meaning to everyone that recognizes the form. For the people who do not know this particular form, “letter a” does not make any sense and similarly for people who do not know the searchlight language, it is just a blinking light, not an “SOS” signal.

Considering symbols as a method of communication, there are diverse forms that we use in our daily life. Many of these symbols go unnoticed on a daily basis because they are universal. Forms do not carry meaning themselves. The human brain then works to recognize the regularities these forms, assign them meanings, and eventually store these meanings in our brains.

We associate form "wrappers" with the real-world meanings which prompt a similar meaning in our brains. On the other hand, two people may give entirely different meaning to the same sentence. What makes them think in different ways even though the input is the same? The answer to this question brings us to Turner and Fauconnier’s “*Mind’s Three I’s*: *Identity, Integration* and *Imagination”*. The answer could be a combination of three things: two people could identify the input differently, integrate the inputs in a different way, or perhaps the new structure that emerged in their brains is dissimilar because their varied background experience. For that reason, *forms* are good but do not explaining everything. There must be another way to explain how we make meanings. Answer to this question comes from cognitive science researchers and linguistics who developed Conceptual Blending Theory.

C. PRINCIPLES OF BLENDING

Conceptual Blending Theory is a complex theory that explains how humans process the information coming from the environment. “Conceptual Blending is a set of operations for combining cognitive models in a network of *mental spaces*.” [4] Mental spaces are the principle entities involved in conceptual blending. In a simplest blend operation, there are three types of mental space:

- Input space
- Generic space
- Blend space[5]

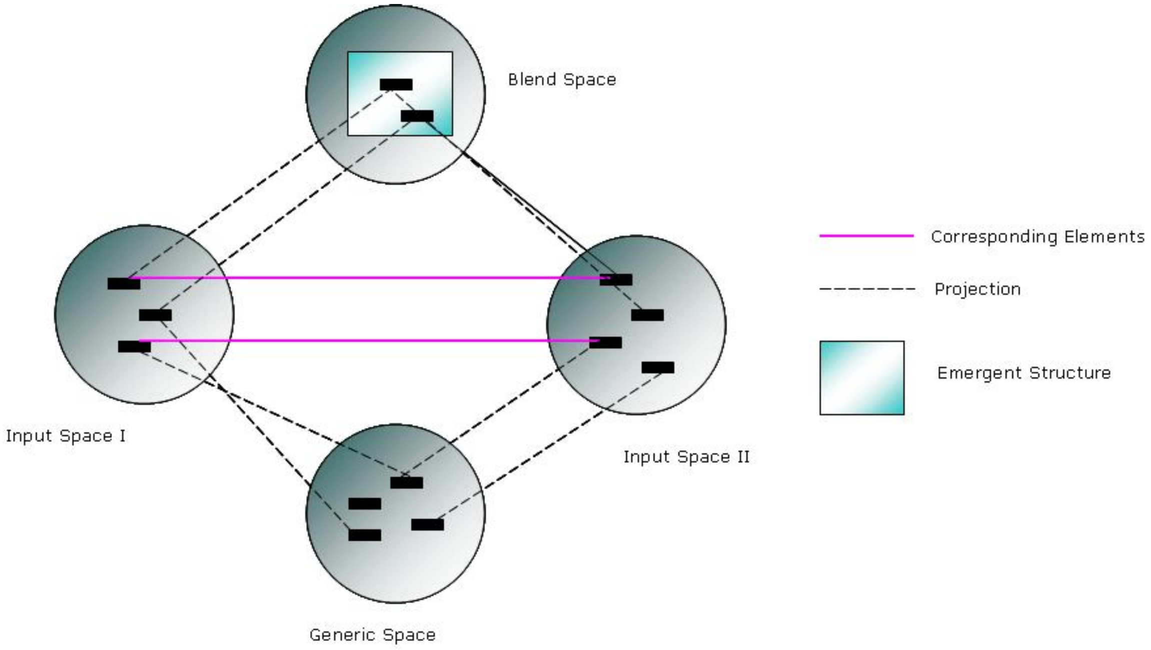


Figure 3. Conceptual Blending (After [1])

Mental spaces are instantly built conceptual containers that appear to be constructed as we talk, listen, remember, imagine and think [6]. Turner and Fauconnier name these containers as “conceptual packets”. Mental spaces contain information about a particular domain. The elements of this information represent entities of whatever we think or any activity we do. They may be related to other elements inside other spaces and may be selectively projected into a "blend space" as shown above. In Figure 3, mental spaces are represented by circles, black rectangles represent elements, lines between elements represent relations between elements, and dashed lines represent projections from one mental space to another.

Blending is an inference method operating on spaces. There may be more than two input space for a blend operation. Generic space contains the common input elements of the input spaces as well as the general rules and templates for the inputs. The elements of generic space can be mapped onto input spaces. Blend space is the place where the emergent structure occurs. The projected elements from each input space and generic space create an emergent structure in the blend space, possibly something not in the input space. The structure in the blend space may be an input for another blend operation as controlled by an Integration network. A new emergent structure may contain not only elements from the input spaces and generic space but also new emergent elements that do not exist in either space [7].

Blending involves three operations:

- Composition
- Completion
- Elaboration

Composition involves relating an element of one input space to another. These relations are called “vital relations”. This matching generally occurs under a "frame". Completion is pattern completion in which generic space is involved in the blending operation. If the elements from both input spaces match the information stored in the generic space, a more sophisticated type of inference can be made, a generalization of reasoning by analogy. This is the place where we use long-term memory and increase our experiences. Elaboration is an operation that creates an emergent structure in the blend space after composition and completion. It is also called running the blend. [8]

“She was so sexy, but he’d heard she was a real cannibal”. [9]

Considering the sentence above, the word "cannibal" is metaphorical and has nothing to do with its original meaning which is “An animal that eats the flesh of other animals of the same kind”. [10] When we read this sentence, we can figure out that the woman who was referenced is probably interested in the man’s money as opposed to his flesh and most likely is using her beauty for this purpose. Disentangling such metaphors and creating newly emergent structure is achieved by the elaboration operation.

We may run blending operations many times for the same input spaces. We cannot often reach a useful blend after one blend operation. We do it subconsciously many times until we find the best result at the end.

> There is always extensive unconscious work in meaning construction, and blending is no different. We may take many parallel attempts to find suitable projections, with only the accepted ones appearing in the final network…. Input formation, projection, completion, and elaboration all go on at the same time, and a lot of conceptual scaffolding goes up that we never see in the final result. Brains always do a lot of work that gets thrown away [11]

Not all elements of the input spaces are projected into blend space. This is called “selective projection”. This is vital to simplify things. Let’s assume that we are looking at a radar scope, following air contacts. There are two input spaces for that case: One for the aircraft values and one for the air-defense concept. The aircraft input space has aircraft’s properties including the kinematics, mission, nationality, type, color. The other input space has the air-defense concept elements. At first we pay attention to aircraft’s kinematics and other relevant factors. We never think about the color of the aircraft even though it is an element of the input space. Identifying the aircraft’s identity does only include other elements of input space but not the color. Therefore the blend space which is the identity of the aircraft does not include the color of the aircraft unless the context requires it. The elements projected onto blend space are selected carefully based on the context in which we are being viewed.

D. NETWORKS OF SPACES IN BLENDING THEORY

Figure 3 is an example of the simplest integration network involving two input spaces, generic space, and blend space. The newly emergent structure may be input space for a further blend operation and linked to another network. A mental space that has been previously used in the network may also be used again in the following blend operations. This gives the model a coherent structure and the ability to explain experience in network form.

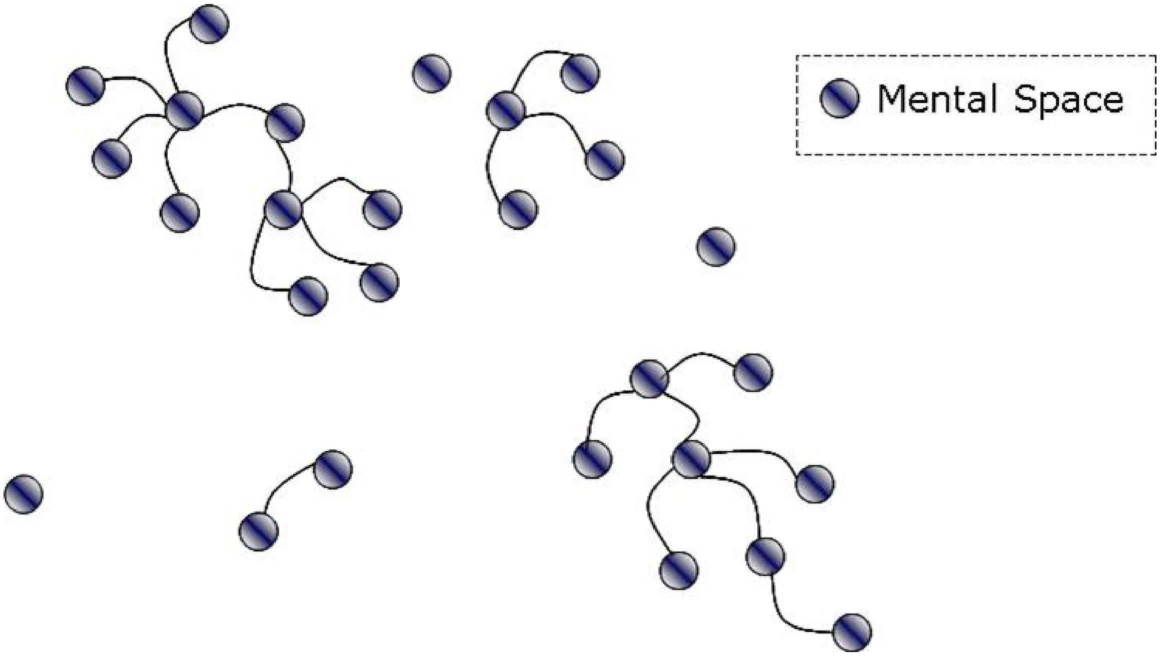


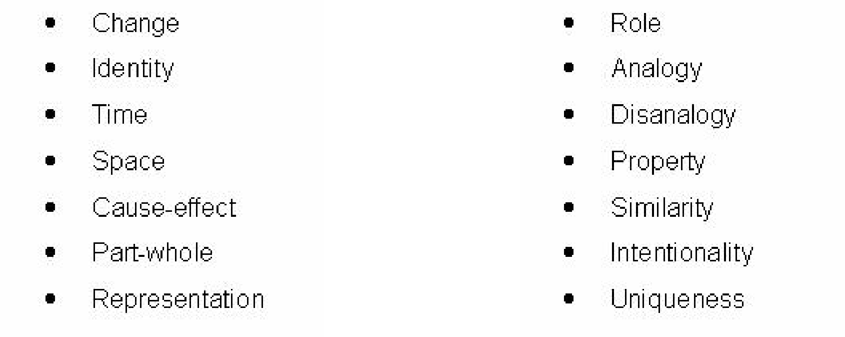
Figure 4. Conceptual Integration Network

An integration network is the focal point of the conceptual blending theory. This network consists of groups of mental spaces and eventually constructs the meaning in our minds by way of blending operations on mental spaces. “An integration network is an array of mental spaces in which the processes of conceptual blending unfold”. [12] The network is constructed by finding mappings between elements in different spaces; projecting these relations from space to space and finally creating an emergent structure that does not exist in either space.

Finding the relations between spaces becomes the most important issue to construct identifiable types of integration networks. At first it seems in Figure 3 that blend space is the most important place of blending theory and therefore makes the blender the most important module of the theory. Actually, this is not completely true. The ability to find the relations between spaces is more important than modeling a blender. These relations are called *vital relations* in blending theory. These relations enable us to combine the two input spaces into one space that is ultimately called *compression*. Turner and Fauconnier define two kinds of relations in blending theory [13]:

- Inner-space relations: relations inside the blend space
- Outer-space relations: vital relations between the input mental spaces

The newly emergent structure is constructed in blend space by compressing outer-space relations into inner-space relations. Turner and Fauconnier have listed vital relations as follows [14]:



- Change
- Identity
- Time
- Space
- Cause-effect
- Part-whole
- Representation
- Role
- Analogy
- Disanalogy
- Property
- Similarity
- Intentionality
- Uniqueness

These vital relations listed above are mostly used in linguistics. However, we can define our own vital relations for an application. Conceptual Blending Theory has four kinds of topology for its integration networks of mental spaces [15]:

- Simplex Network
- Mirror Networks
- Single-Scope Networks
- Double-Scope Networks

The type of topology is primarily related to the organizing frames. The similarity of the organizing frames in each input space determines the type of the topology. Organizing frames may or may not be the same for both input spaces. If they are not the same, clashes occur between input spaces. One of the organizing frames or a hybrid of the two frames determines the frame of the blend space.

1. Simplex Networks

Simplex networks have an "organizing frame" in one input space and relevant data in the other input space. These networks are good at variable-value type of relations. If we have a “track info” organizing frame in one input, the speed, heading, location, identity and other variables are represented by the elements in one input space. In the other input space there are values for each element in the organizing frame. In Figure 5, input space I has the data for the speed variable for input space II. Primarily in this kind of network role-value relations are used. Simplex networks basically formalize first-order logic proofs as studied in artificial intelligence.

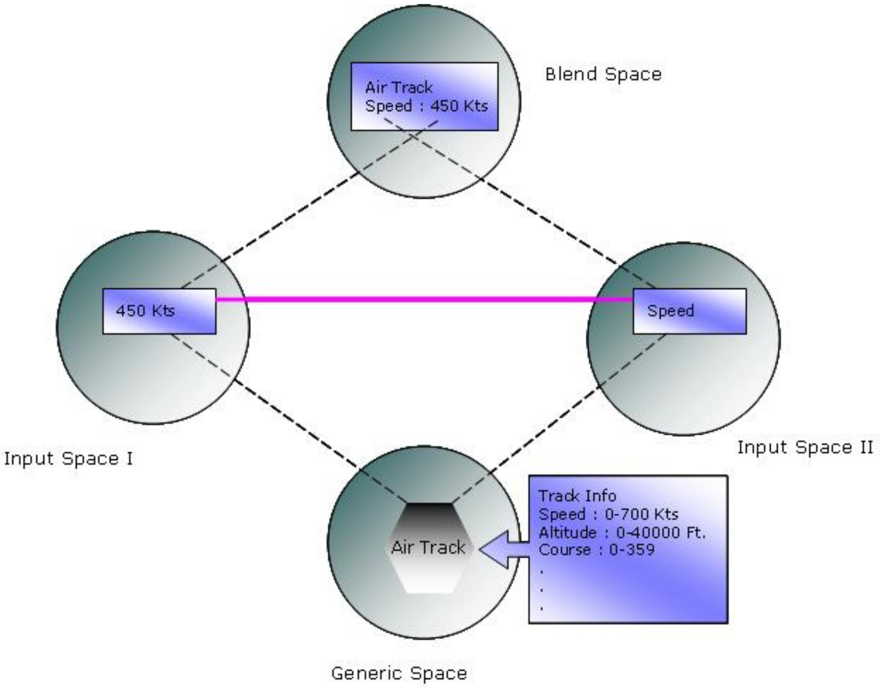


Figure 5. Simplex Network (After [1])

2. Mirror Networks

In mirror networks, both of the mental spaces, generic space and blend space, share the same organizing frame. Since both input spaces share same organizing frame finding relations between inputs are straightforward. Therefore there is no clash between mental spaces in the blending at the level of organizing frame. However there may be clashes between subframes of organizing frames. [16]

Turner and Fauconnier explain mirror networks with a comparison of the cruise time of two sailing ships leaving San Francisco for Boston. In 1853 the clipper ship named *Northern Light* made this voyage in 76 days 8 hours and this was a record time until another modern catamaran named *Great America II* made this distance in shorter time in 1993.

A few days before the catamaran reached the Boston, the observers were able to say that Great America II is 4.5 days ahead of Northern Light [17].

This sentence discusses two boats racing with each other and one of them is 4.5 days ahead of the other one. However, these two boats are not competing with each other and they don’t even exist in the same time period. When we read this sentence, we can understand that Northern Light was in the analogous position in 1853 but 4.5 days later than Great America was. One of the inputs to that blend operation is Northern Light cruising in 1853 and other one is Great America in 1993. The organizing frame in the blend operation is sailing a boat from San Francisco to Boston. Only boats, time periods and position on course are projected on to blend space while weather conditions and the aim of voyage are not. Time vital relation enables us to associate these two events in the same time domain by compressing time. Compression is evaluating two events with 140 years time difference in the same space and seeing them as if happening at the same time.

There is no clash between the organizing frames of these two events. On the other hand, there are some clashes in the subframes. While one input has a nineteenth century cargo sailing boat, the other input has a twentieth century racing catamaran. Both frames clash each other but these clashing properties are heuristically ignored in the blend. [18]

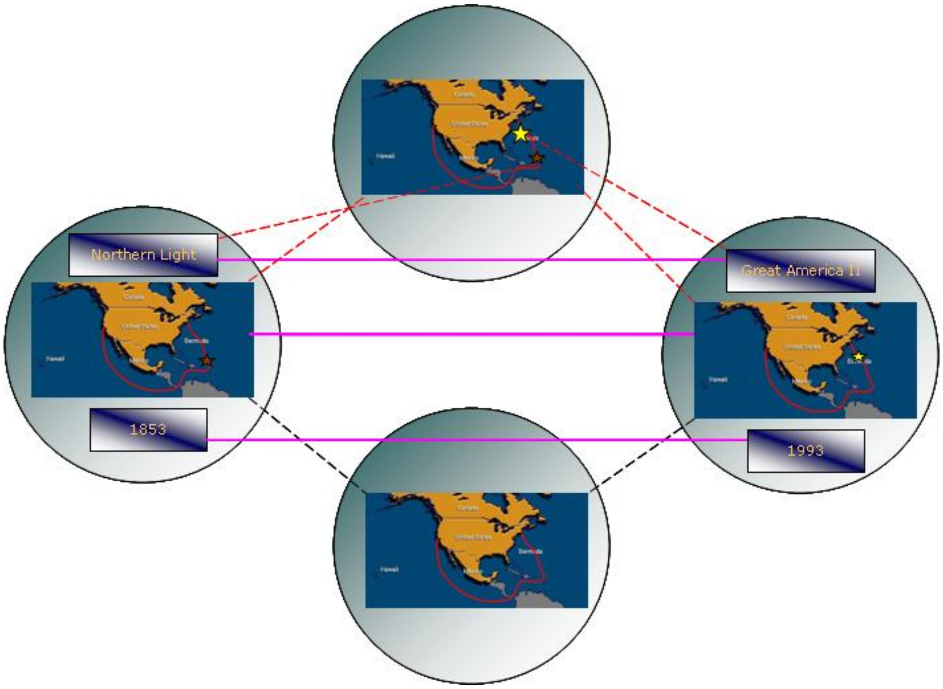


Figure 6. Mirror Network (After [1])

Besides simplex and mirror networks, there are also single-scope networks and double-scope networks in blending theory. In single-scope networks the two input spaces have different kinds of organizing frames and only one of them shows up on the blend space. In double-scope networks the input spaces have two different organizing frames again but this time a combination of these frames show up in the blend space.

E. NETWORKS IN THE ADL SIMULATION

The ADL Simulation uses only the simplex and mirror networks. Simplex networks are used in assigning vales to variables of air tracks. At the reactive level each reactive agent receives its information and triggers another action at the predictor track-agent level.

Mirror networks are used in various places in the ADL Simulation. They are used in "regional" space to figure out coordinated activities between tracks. In a coordinated attack scenario, two air tracks are turning inbound at the same time. Both of the input space organizing frames are the same: Attacking a ship. We can compress these two input spaces with the place and time vital relations and find a coordinated attack profile and then project this conclusion onto blend space.

Another place that mirror networks are used in the ADL Simulation is in the track-agent level. Consider an air track that is changing its course. We can find out the change in course by comparing two heading values in successive times. Two input spaces have two different heading values in two different times. The organizing frames are same but the time elements of the input spaces are different.

F. ALTERNATIVE WAYS TO BLENDING THEORY AND CMAS LIBRARY FOR ADL SIMULATION

Using the CMAS library is not the only way of coordination and communication to implement ADL Simulation. An alternative to usage of CMAS library is to pass arguments to each other and define methods for each of communication links. This may be good for an environment where there are many agents. In a mesh topology in where n agents have dedicated point-to-point links to every other, we need n(n-1)/2 links. For a limited number of agents in the environment this number may be acceptable but in ADL Simulation's layered structure there are more than 15 agents for each track. For a multithreaded environment this number will multiply itself for dedicated links between every agent. The requirement of links and coordination between each agent could be a considerable advantage of the usage of the CMAS library. Therefore we used CMAS library for coordination and communication between agents instead of using a mesh topology and dedicated link between each agent.

THIS PAGE INTENTIONALLY LEFT BLANK

III. RELATED WORK IN NAVAL AIR-THREAT ASSESSMENT

A. RELATED WORK INTRODUCTION

We reviewed a variety of previous work that relates to the subject of naval air-defense, cognitive modeling, threat assessment and how the human brain works. Many studies were conducted after the *USS Vincennes* and the *USS Stark* incidents to understand the underlying reasons and the factors affecting decision-making under stress. While most studies focused on increasing the accuracy of decisions made under stress and the performance of watchstanders in the Combat Information Center on board ships, there has not been much study on a cognitive model for the human contribution to decisionmaking. A few studies of how humans do identification and threat assessment suggests that humans get inputs from environment, compare them with some predefined templates, and then make a decision.

B. ADVERSARIAL PLAN RECOGNITION FOR AIRBORNE THREATS

A plan recognition system for airborne threats was developed by Richard Amori, the Plan Recognition for Airborne Threats (PRAT).[19] PRAT performs three-dimensional spatial and temporal reasoning, incorporating a high volume of data with predefined patterns via two different kinds of agents. The PRAT system used the Falkland war between Argentina and Great Britain in its scenarios.

The most important module of the system is the Plan Recognizer which is an intelligent subsystem. This module is based on physical data and changes to this data, known air tactics and behaviors, and likely primary and secondary goals. This module has two subcomponents: the Individual-Agent Manager and Sets-of-Agents Manager. The former analyzes the data associated with each track while the latter analyzes the coordinated activity between air threats. These two modules use each other’s inferences so that both can provide mutual support for more accurate reasoning. Each of these modules uses "rolling" data structures that evolve and are updated continuously. Each data structure has forward and backward components. A backward component is supplied with incoming data from the environment while a forward component provides the reasoning about the track or tracks. “A backward component permits reasoning which is historical in nature, and the forward component permits reasoning which is hypothetical in nature”. [19]

In naval air warfare, a large amount of data needs to be processed. In high-threat situations this becomes even more severe. The PRAT system addresses this problem by dividing responsibilities into sets of agents. Each module is also subdivided into several different mission reasoners.

The PRAT system is similar to the ADL Simulation in two ways. First, both use a layered agent architecture rather than using one type of agent. Secondly, both use some common factors to identify track intention and identity like kinematics values. One difference between ADL Simulation and PRAT system is that the PRAT system uses only kinematics values while the ADL Simulation uses additional factors. The ADL Simulation also uses special CMAS data structures instead of keeping data in a data structure and traversing this data each time to find a match or reason about existing history data as in the case of PRAT.

C. NAVAL AIR-DEFENSE THREAT ASSESSMENT: COGNITIVE FACTORS AND MODEL

Another investigation examined the cognitive aspects of naval air-threat Assessment. Experienced US Navy Air-defense personnel were used in this research. Collected data revealed that participants assigned threat and priority levels to air tracks by using a set of factors.

> Factors are elements of data and information that are used to assess air contacts. Traditionally, they are derived from kinematics, tactical, and other data. Examples of such data include course, speed, IFF modes, and type of radar emitter. [20]

The major factors were electromagnetic signal emissions, course, speed, and altitude, point of origin, Identification Friend or Foe (IFF) values, intelligence, airlane, and distance from the detector. Participants used up to 22 factors, and each used different but overlapping factors. Participants used more factors in identifying tracks posing a greater threat than tracks posing a lesser threat. Participants used the factors in a certain order in threat assessment and each factor had a priority.

Each factor has an expected range of values. Research showed that aircraft that matched these expectations were assigned lower threat levels than aircraft that did not. Figure 7 shows a threat assessment model derived from the research.

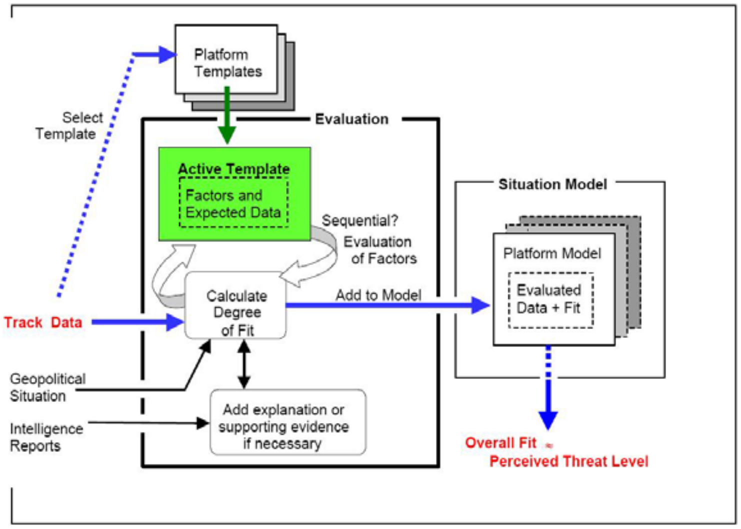


Figure 7. Cognitively-Based Model of Threat Assessment (From [20])

The ADL Simulation uses the factors found in this research in its reactive agents in the threat assessment process. The ADL Simulation improved on this model by adding an integration network and the ability to evaluate coordinated activities between two or more aircraft.

D. AIR-THREAT ASSESSMENT: RESEARCH, MODEL AND DISPLAY GUIDELINES

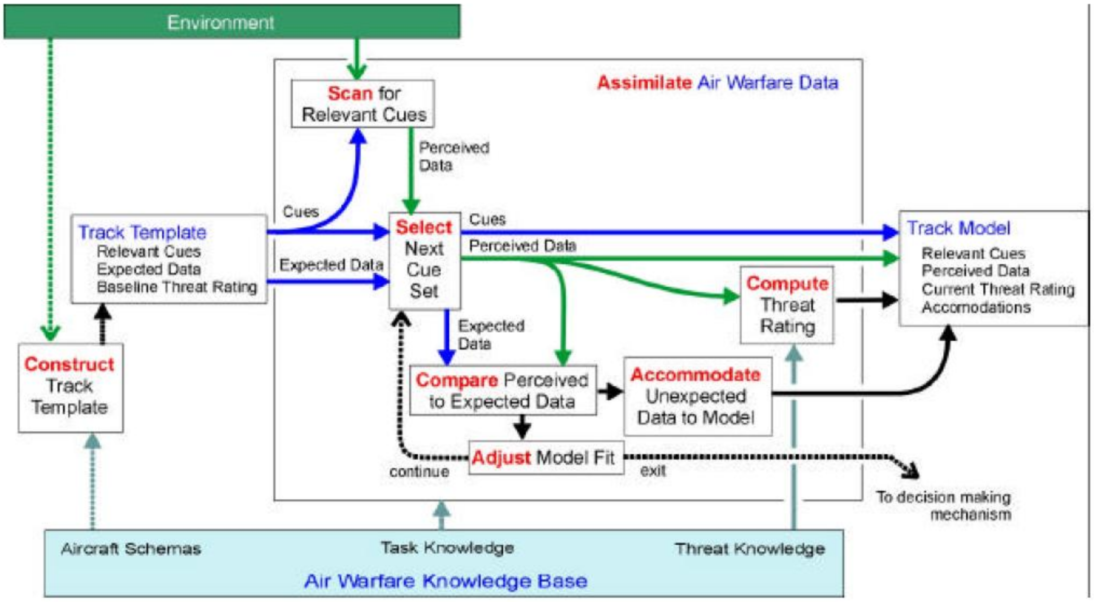


Figure 8. Threat Assessment Model (From [22])

The factors in the previous model are mentioned as cues in another study. Each cue has a weight and if the perceived data is unexpected, the value of the active model is reduced by the weight of the clue. Other studies discussed in this paper includes Tactical Decision Making Under Stress (TADMUS), the Decision Support System (DSS), and the Basis For Assessment (BFA) concerning the development of more efficient tactical displays and human interfaces for air-defense personnel. These studies showed that if the most important data is shown to the user more effectively, accuracy is increased in threat assessment.

The following results were found in the research:

- Users created templates to define which cues will be evaluated and the permissible range of data for each cue.
- Cues were:
    - Evaluated in a fairly consistent order;

    - Weighted;
    - Processed in sets reflecting their weights.
- Air-defense threat evaluators:
    - Did not rely on all data, only the data associated with cues in their active template;
    - Did not change templates in the face of conflicting data;
    - Were influenced by conflicting data in specific cues rather than in the overall pattern.
- Perceived threat level:
    - Was related to the degree of fit of observed data to expected data ranges in the evaluator's active template;
    - Was not related to the number of cues that were evaluated during threat assessment [23].

E. SIMULATION OF AN AEGIS CRUISER COMBAT INFORMATION CENTER

Other previous work reports on a simulation that models the CIC of an Aegis Cruiser for air defense. The research mainly explores the team's performance under high-stress situations and tries to understand the interpersonal factors that affect the overall performance of the CIC team and watchstanders. Air-defense contact identification, threat assessment, and classification were modeled but were not the primary focus of the research. An artificial neuron is used to model the cognitive decision-making process.

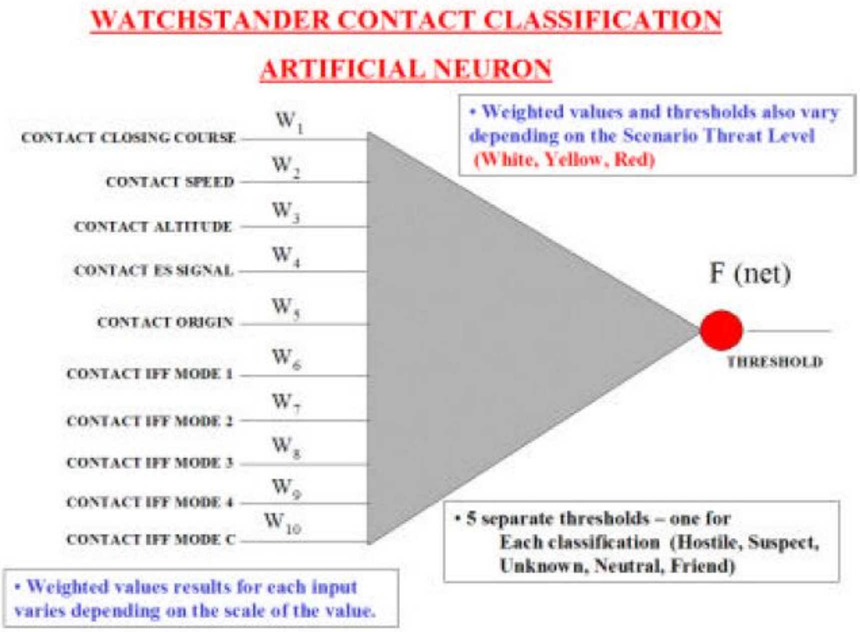
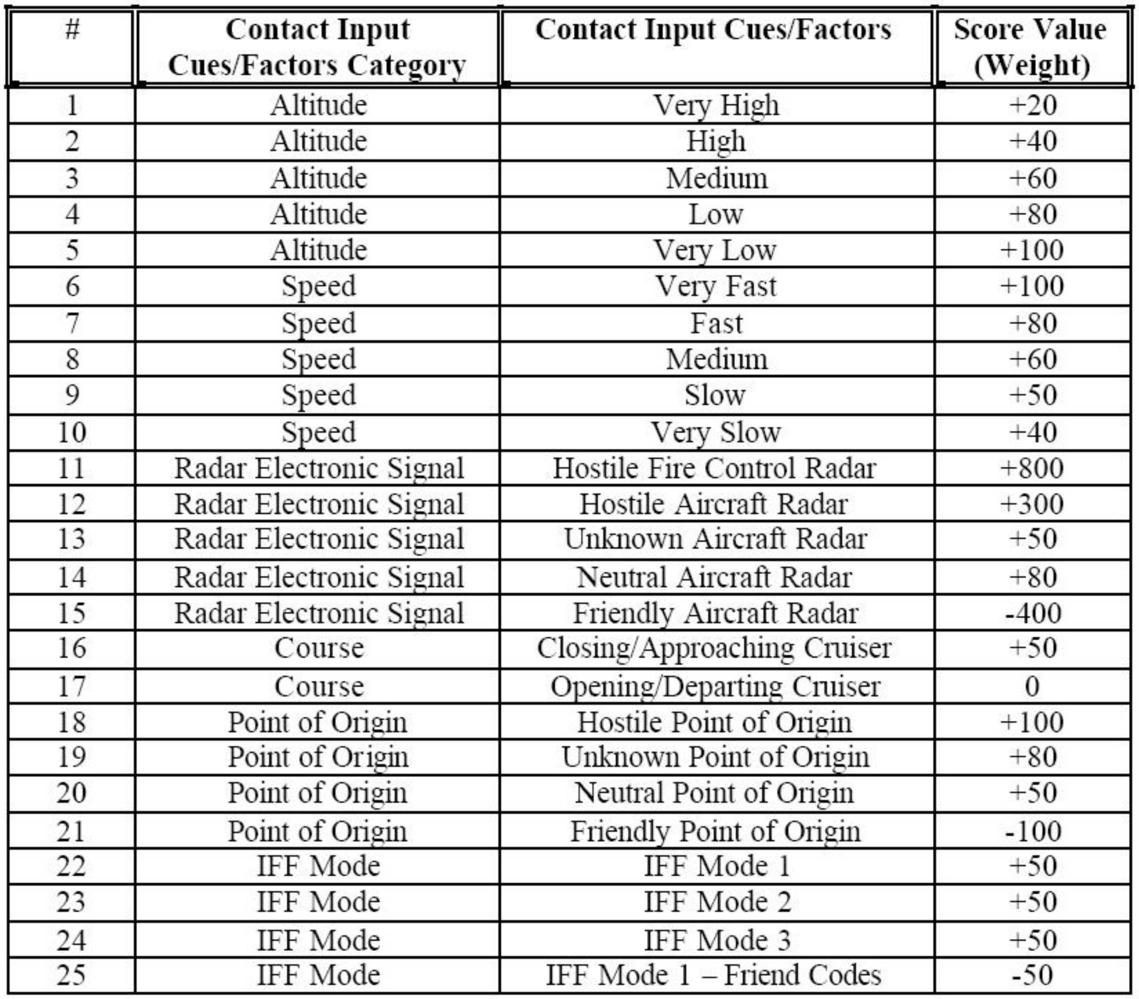

WATCHSTANDER CONTACT CLASSIFICATION
ARTIFICIAL NEURON
• Weighted values and thresholds also vary depending on the Scenario Threat Level (White, Yellow, Red)
CONTACT CLOSING COURSE
CONTACT SPEED
CONTACT ALTITUDE
CONTACT ES SIGNAL
CONTACT ORIGIN
CONTACT IFF MODE 1
CONTACT IFF MODE 2
CONTACT IFF MODE 3
CONTACT IFF MODE 4
CONTACT IFF MODE C
W_1
W_2
W_3
W_4
W_5
W_6
W_7
W_8
W_9
W_{10}
F (net)
THRESHOLD
• 5 separate thresholds – one for Each classification (Hostile, Suspect, Unknown, Neutral, Friend)
• Weighted values results for each input varies depending on the scale of the value.

Figure 9. Contact Classification Artificial Neuron (From [24])

Each contact category has a threshold value for classification and threat level. The threat level may be White, Yellow or Red. Each input value has a weight and weighted sum is compared to a threshold. The scoring values and thresholds were constructed in compliance with air-defense personnel from the ATRC detachment in San Diego, CA. The threshold values are displayed below:

Contact Classification	Threat Level White Thresholds	Threat Level Yellow Thresholds	Threat Level Red Thresholds
Hostile	= +600	= +500	= +450
Suspect	500 – +599	+450 – +499	+400 – +449
Neutral	400 – +499	+300 – +449	+200 – +399
Unknown	-399 – +399	-399 – +301	-399 – 199
Friend	= -400	= -400	= -400

Table 1. Default Classification Threshold Values(From [24])



#	Contact Input Cues/Factors Category	Contact Input Cues/Factors	Score Value (Weight)
1	Altitude	Very High	+20
2	Altitude	High	+40
3	Altitude	Medium	+60
4	Altitude	Low	+80
5	Altitude	Very Low	+100
6	Speed	Very Fast	+100
7	Speed	Fast	+80
8	Speed	Medium	+60
9	Speed	Slow	+50
10	Speed	Very Slow	+40
11	Radar Electronic Signal	Hostile Fire Control Radar	+800
12	Radar Electronic Signal	Hostile Aircraft Radar	+300
13	Radar Electronic Signal	Unknown Aircraft Radar	+50
14	Radar Electronic Signal	Neutral Aircraft Radar	+80
15	Radar Electronic Signal	Friendly Aircraft Radar	-400
16	Course	Closing/Approaching Cruiser	+50
17	Course	Opening/Departing Cruiser	0
18	Point of Origin	Hostile Point of Origin	+100
19	Point of Origin	Unknown Point of Origin	+80
20	Point of Origin	Neutral Point of Origin	+50
21	Point of Origin	Friendly Point of Origin	-100
22	IFF Mode	IFF Mode 1	+50
23	IFF Mode	IFF Mode 2	+50
24	IFF Mode	IFF Mode 3	+50
25	IFF Mode	IFF Mode 1 – Friend Codes	-50

Table 2. Scoring (Weighted) Values for Various Input Cues(From [24])

F. MULTISENSOR DATA FUSION

Another relevant study discusses data-fusion techniques, collecting data from multiple sources and combining them to achieve more accurate results than could be achieved from a single sensor alone. Data fusion has military applications (e.g. finding track identity and establishing a tactical picture) and non-military applications (e.g. robotics, automated control of smart buildings, weather monitoring, and medical applications). Air-threat assessment involves data fusion since air-defense personnel have to combine information from multiple sensors and evaluate them.

> To establish target identity, a transformation must be made between observed target attributes and a labeled identity. Methods for identity estimation involve pattern recognition techniques based on clustering algorithms, neural networks, or decision-based methods such as Bayesian inference, Dempster-Shafer's method, or weighted decision techniques. Finally, the interpretation of the target's intent entails automated reasoning using implicit information, via knowledge-based methods such as rule-based reasoning systems [25].

The fusion in the ADL Simulation used a combination of a neural network in the form of an integration network and an evidence weighting algorithm. An integration network is used by the Conceptual Blending Theory. The models that reside in the nodes of this network are weighted based on some Bayesian inferences.

This study also suggests Blackboard Knowledge-Based Systems (KBS) as a good data-fusion method. These systems partition the problem into subproblems and use constrained interaction between solutions of subproblems to solve whole problem. This is analogous to how human experts come to a solution by gathering in front of a blackboard and brainstorming. A KBS must have three required elements:

a. Knowledge representation schemas

b. An automated inference/evaluation process

c. Control schemas

The first and third requirements are provided by Generic Space while the second requirement is done by the blender in Conceptual Blending Theory. Thus the method used in ADL Simulation (Conceptual Blending Theory) fulfills the requirements of a KBS. At the same time, the ADL Simulation is a cognitive model for how humans accomplish these functions.

G. MULTISENSOR DATA FUSION

Another study about multi-sensor data fusion distinguishes three kinds of data fusion: data, feature, and decision. For data fusion, raw data from each sensor is combined in a centralized manner. This is claimed to compute the most accurately of the three. The drawback of this method is the requirement that all sensor values must be put in the same units. If sensors are distributed in the real world, all the information from all of the sensors must be transmitted to the center, which requires a large bandwidth. In feature-level fusion, features are extracted in each sensor and these features are transmitted to the center. In this case communication requirements are reduced but the result is less accurate because of the lost information during generating features from raw data. Finally, in decision-level fusion, each sensor sends a decision about its input and these decisions are fused. The result is the least accurate of the three fusion options because of the information compression of the sensor observations, but requires the least bandwidth.

In ADL Simulation, predictor track agents are like local sensors focused on individual aircraft tracks. They are using feature-level fusion in which sensors are represented as reactive agents. Predictor agents thus infer the identity of the aircraft based on the features sent by the reactive agents. At the same time, regional agents identify coordinated activities between aircraft and collect data from track agents, doing something like decision fusion. Thus, different levels of data fusion are used in the ADL Simulation.

H. COMPARISON WITH OTHER AIR-DEFENSE WORK

In summary, the ADL Simulation is unique for the following reasons:

- It uses Conceptual Blending Theory to imitate a human brain.
- ADL Simulation uses a Connector Based Multi-Agent System to create an integration network.
- The ADL Simulation allows a user to set up an arbitrary geographical area to test.
- It is structured with a layered design.

- The ADL Simulation can use both analog and digital approaches to inference to permit studying the precision losses that come with digitization.
- Its results are stored in an XML file which enables studying this data.
- It allows a user to see the decisionmaking process in a step-by-step manner and give reasons for decisions. A user may see all decision-making processes by sending queries to a track.

IV. MULTI-AGENT SYSTEMS AND DESCRIPTION OF AIR DEFENSE LABORATORY SIMULATION

A. PROGRAM LANGUAGE AND SYSTEM REQUIREMENTS FOR ADL SIMULATION

The ADL Simulation is written in the Java programming language. It was developed using the JBuilder 9 Application Development Environment. The Java Development Kit 1.4.1_02 is used for the implementation of the program. The program was run on a Pentium 4, 2.4 GHz. machine with 512 megabytes (MB) of RAM. The requirements to run the program are as follows:

- Pentium 3 or equivalent and higher processor
- Minimum 512 MB of RAM
- Java Development Kit 1.4.1 or higher
- Screen display of 1280x1024 pixels or higher

The program is based on a multi-threaded environment: There are more than 100 threads running in a five-track scenario. Therefore a processor with high speed and large amount of RAM is a requirement for the program to run smoothly. The SPY XML Editor was used to monitor data logging information and XSLT transformations. XML files are used to store data logging information.

B. MULTI-AGENT SYSTEMS AND THE CMAS LIBRARY

1. Agents

A multi-agent system (MAS) is a computing model made of entities that communicate with each other and act in an environment. [2] Agents are autonomous software elements operating in an environment. Multi-agent systems have six components: an environment (E) which is the space where agents operate, the objects (O) in the environment, actors (A), relations (R) between actors and the environment and relations between agents, operations (Ops) that are executed by actors, and laws of the environment.

Agents in a multi-agent system environment receive input from the environment, process this input, and produce an output. Then the agents release the output back to the environment. This kind of architecture may be basically formulated as sense-process-act.

According to Integrated Asymmetric Goal Organization (IAGO) team at NPS there are three kinds of agents: reactive, cognitive, and composite. The actions taken by reactive agents only rely on the current input data, and these agents do not use memory or experience. Therefore there is no learning capability in reactive agents. They are good at basic implementations (e.g. thermostats or alarms).

Cognitive agents maintain a state of information and knowledge which permits them to operate in conjunction with the memories and experience gained so far. Composite agents are composed of both reactive agents and cognitive agents, typically in a hierarchy. Such agents communicate with the inner environment of the host agent as well as the outer environment. Inner agents maintain an insight model and internal states for host agent.

An alternative taxonomy gives four kinds of agents: simple reflex, environment trackers, goal-based, and utility-based. Simple reflex agents are associated with the definition of reactive agents. Agents that keep track of the environment have some sort of state information but they are not quite cognitive enough. Goal-based agents address certain kinds of goals. Utility-based agents try to make agents happy on the way to the goal. While goal-based agents use only one path to a goal, utility-based agents use the most effective path.

2. Connector Based Multi-Agent Systems and CMAS Library

a. Connectors

Connectors are one way to do communication and coordination between agents. They are a particular kind of message passing system. [26][27] Only the agents with the same namespace of the source agent may receive the data through the connectors. Naming the connectors with a namespace provides an addressing facility in the communication between agents. There are three different states for a connector: retracted, extended, and matched.

Retracted connectors are the ones that are not broadcast by agents. They cannot be matched to another agent. Such a connector may be retracted because the connector could not match in a certain time, the conditions that created the connector have changed, or the conditions have not yet been met to extend the connector. In an extended connector, the inner state information is made available to the outside environment for another agent with the same kind of connector to match. Matching connectors may fire an action or change the state of a data structure in an agent.

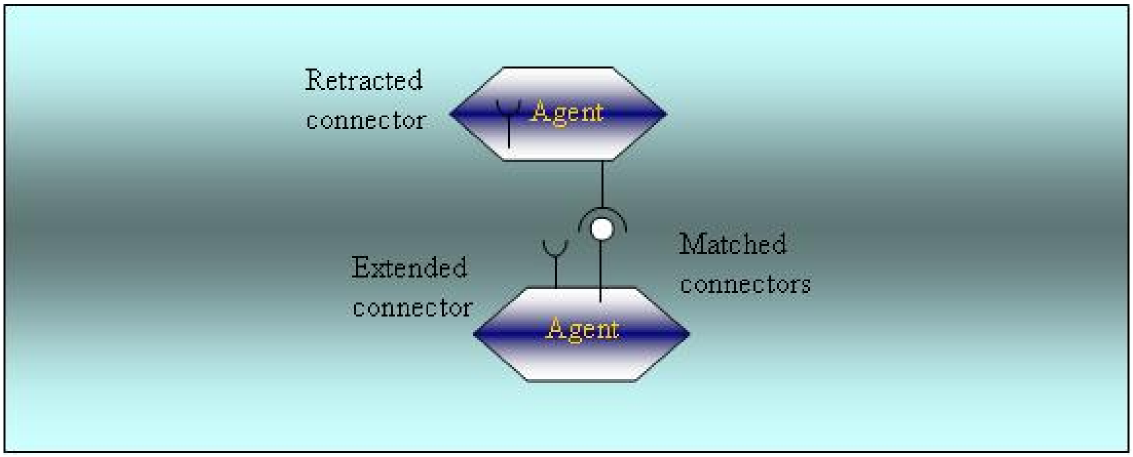


Figure 10. Connector States

One could question that there are other many ways proposed for agents to communicate. A general message passing could be used to communicate the agents. During the research before we received the library we used a general message passing method. As the project got bigger the amount of code also increased. After we received the CMAS library, we used connectors to communicate and coordinate the agents inside the simulation. With the CMAS library, we decreased the amount of code for coordination and communication of agents. This enabled us to focus on the model rather than the coordination of agents. For our second goal, use of the CMAS library enables us to extend the project further in the future.

b. Tickets

Tickets are procedural instructions for agents and data organizing systems. There are basically two kinds of tickets: data and procedural. Data tickets organize the data structures inside an agent. These tickets have different kinds of frames. These frames may include names, types, and type-value pairs. The status of a ticket is determined by the status of each frame inside the ticket. Data tickets may be used as a trigger to fire an action when a set of data is matched with predefined criteria. Procedural tickets have methods to be executed. When certain conditions are met the methods in these frames are executed in the ticket.

Both data tickets and procedural tickets have two states: completed and incomplete. A data ticket may be completed when either all of the frames in the ticket are "set" or a predefined subset of all frames is "set". Procedural tickets may be completed when either all of the frames are executed or a subset of them is executed. The frames that have to be set or executed to make a ticket complete are called primary frames.

A ticket may also be sequential or non-sequential. Sequential ticket frames must be completed in sequence while frames may be executed or set out of order in a non-sequential ticket. In a sequential ticket, each frame may set or fire the other frame to set or execute. The tickets in which the all frames have to be set or executed without interruption are called synchronous tickets, or must-complete tickets. Tickets in which the frames may be set or executed in any time are called asynchronous tickets.

There are other ways to associate procedural and logical information with an agent. Plans, rules, and scripts are some of them. The difference between them and tickets are that tickets are more abstract. We can do more things with tickets. Tickets retain the plans, scripts, rules, and data structure of an agent. Most of the implementation of tickets in CMAS library is interfaces not classes. Tickets of the library are left to user to be implemented based on the context.

c. Ticket & Connector Structures

Tickets and connectors may be used together to achieve various kinds of design options and complete coordination between agents. Connectors may be used to activate a ticket. This kind of relation may be a precondition for the ticket. After some certain conditions are provided tickets are activated and the frames in the ticket are executed.

Connectors may be used as a trigger for methods implemented in procedural tickets or to set data structures in data tickets. Then connectors are gates to individual frames inside the ticket. Once a connector is matched with another one, methods in frames in a ticket may be fired. In that sense connector matching acts like a function call. An action taken by a frame inside a ticket may be a trigger for a different frame inside another ticket. Then output which is released from a frame of a ticket may be connected to an additional frame inside another ticket.

Many MASs are nested agent systems. At the very bottom level, reactive agents are the working units. Above them is there another agent system with more cognitive agents and so forth. Each agent system makes their own decisions in their local area called a "membrane" or context. The relations with upper and lower level membranes are provided by connectors. These connectors enable systems to be generalized. With generalization, agent systems can not only affect their little environment but also affect the outside environments and receive information from outside world. This is called feedback.

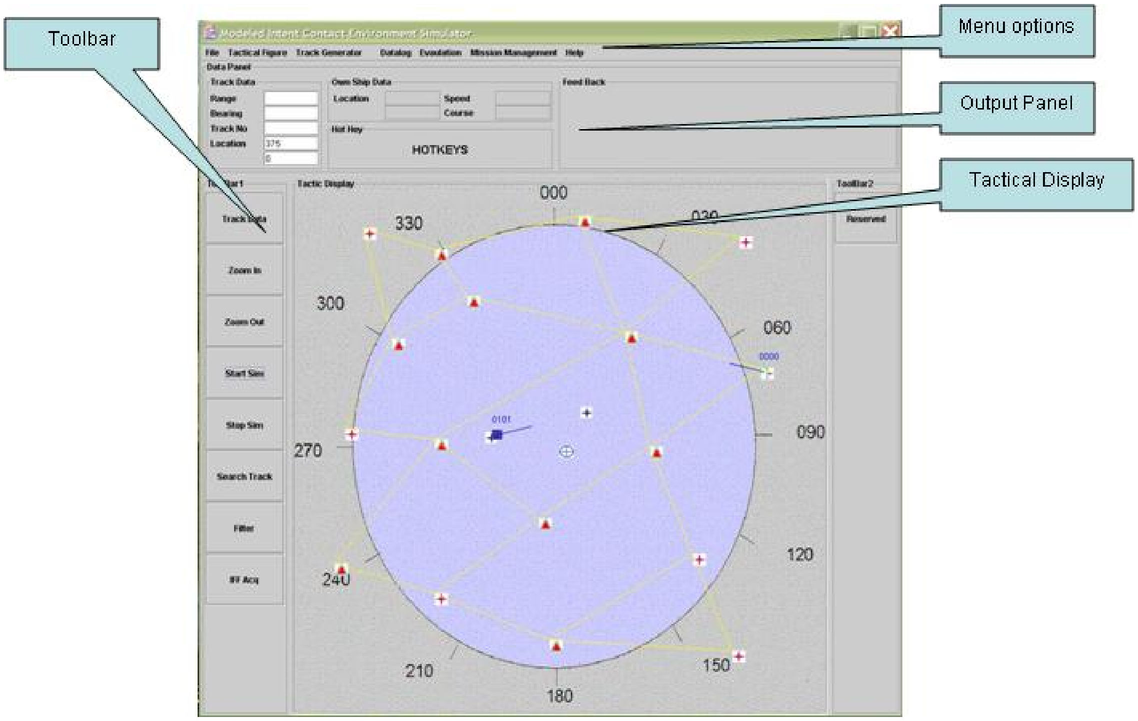
d. The CMAS Library

CMAS library was written by Neal Elzanga for the IAGO at the Naval Postgraduate School. This library enables users to define five types of connectors: String, Integer, Double, Float, and Boolean connectors. Each connector must be given a namespace to enable matches in the CMAS library. These namespaces stands for membrane. If the connectors are registered to CMAS library, any query with registered namespace in the software can reach the value of the connector if the connector status is extended. That enables the user to communicate the agents between each other without implementing an external connection inside the software. CMAS library also enables the user to define tickets. Both data and procedural tickets are implemented in the CMAS library. The library and IAGO project is still on progress at Naval Postgraduate School.

C. MENU OPTIONS

The ADL Simulation user interface has four main components as shown in Figure 11: menu options, output panel, toolbar, and tactical display.

The tactical figure has four drop-down menus. These menu options allow a user to specify the environment, create and delete airbases, air routes, and joint points, create user-derived aircraft, load a pre-prepared scenario, save a prepared scenario, and create an Air Tasking Order (ATO) message for friendly activity in the environment. ATO Messages are prepared by the Air Force or Naval Force holding air assets in their force on daily basis, and this message informs all friendly forces of the friendly air activity in the area with time frame, task area, IFF values and mission specifications.



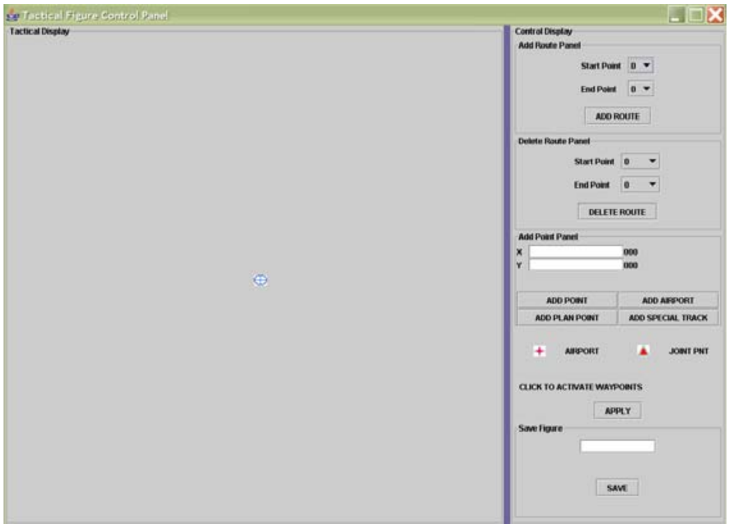
Location 375
0

Figure 11. Visual Design of the ADL Simulation

One submenu opens a new window for the user to define the tactical scenario for the simulation. The user can create an environment by adding airports, joint points, and air routes between these points (Figure 12). The simulation finds the shortest path from each airport to every possible airport and stores them as waypoints. The method used to find the shortest route is a combination of the A* and Depth First Search (DFS).

The user can also create a track by specifying the waypoints on this panel. The user can specify the altitude, speed, IFF Transponder status, IFF values, radar status, and radar emission on this panel. Agents controlling the actions of the aircraft adjust the altitude and speed of the aircraft based on the values on the waypoints. IFF and radar status changes on the waypoints are based on the geographic location of the aircraft.

Another menu option loads the default scenario from the hard disk. This scenario includes the location of airports, joint points and the routes connecting them. This enables the user to test the same scenario multiple times without recreating the same environment. The user can also create an ATO message.



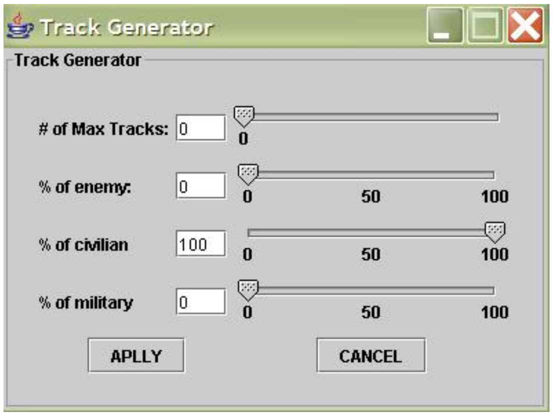
Start Point 0
End Point 0

Start Point 0
End Point 0

X 000
Y 000

Figure 12. Tactical Figure Control Panel

There are four submenus under the Track Generator: Generate Tracks, Add Coordinated Attack with Snooper, Add Coordinated Detachment Attack, and Add Missile Attack. The user can define the maximum number of randomly generated tracks in the environment. User derived tracks, the snooper, coordinated detachment attack track, and missile attack track are not included in this number. The user can also define the percentages of the types of aircraft on this panel.

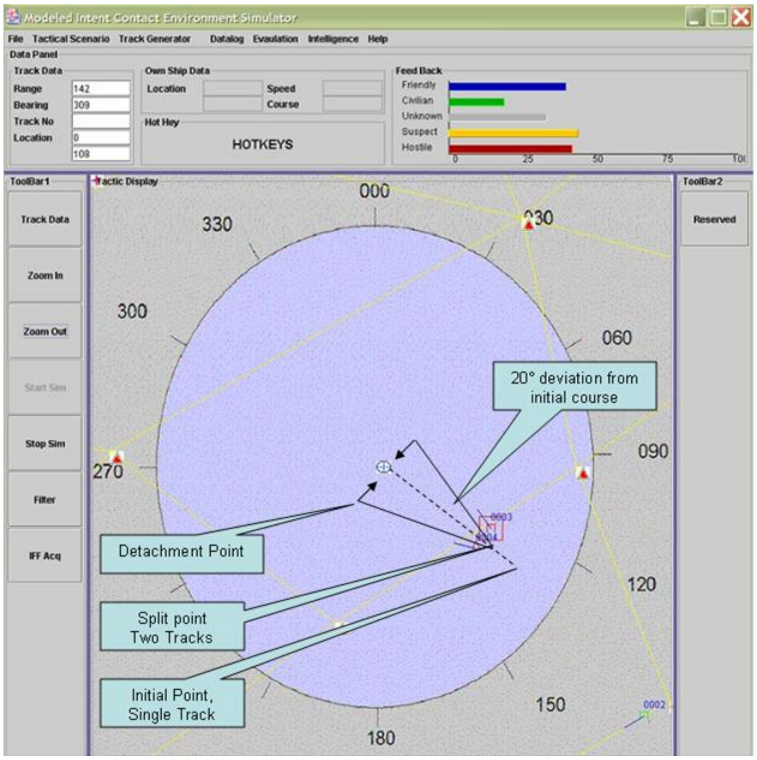


# of Max Tracks: 0
% of enemy: 0
% of civilian 100
% of military 0

Figure 13. Track Generator Panel

The snooper is an opponent aircraft charged with collecting intelligence in a given area. These aircraft orbit at a specific location and they stay out of effective weapon range of the ships to protect themselves from surface engagements. They pinpoint the location of surface contacts for striker aircraft. The snooper track is an actor of one of the coordinated attack scenarios of the simulation. The location and behavior information of the snooper is loaded into the simulation when this particular option is selected.

An enemy aircraft can also be created in an expected threat sector. When the distance is about 30 nm, this track is split and another track shows up on the screen. They change their direction 20° away from the initial course but each one in a different direction. When the surface ship is on their beam, both aircraft return inbound and attack the ship. After the engagement is completed both tracks fly away from the ship and merge again. The scenario figure is shown below.



Range 142
Bearing 309
Track No
Location 0
108

Figure 14. Coordinated Detachment Attack

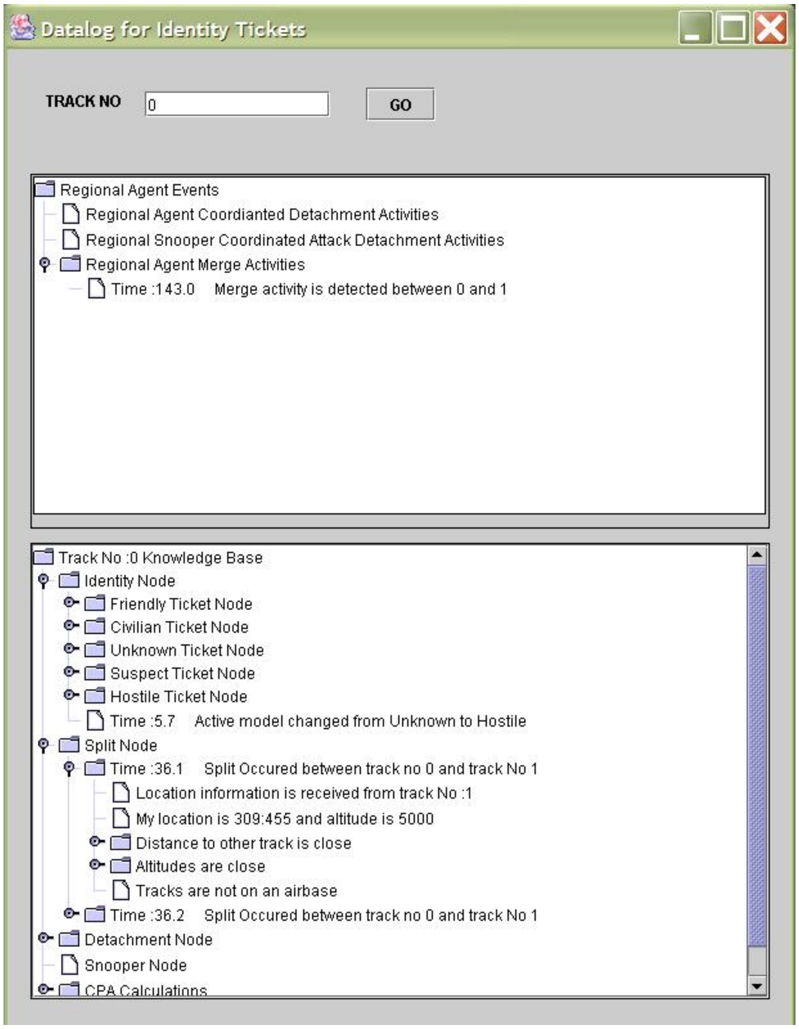
An enemy aircraft loaded with Air-to-Surface Missile (ASM) can be created in the expected-threat sector. It directly approaches ships until its range is about 30 nm. Then the aircraft releases its missile and turns away from the ship. This action is simulated in the ADL Simulation by splitting the air track.

The user can choose to turn the datalog option on or off and define the datalog frequency to store track agent information to an XML file. The ADL Simulation periodically stores all the information to an XML file if the datalog is turned on. The Analog & Digital selection panel enables the user to select the criteria for the evaluation process at the predictor-agent level and reactive-agent levels

The evaluation of model panel shows the integration network created by the ADL Simulation for the regional agent and the selected track. The integration network is represented in a tree structure. The user can traverse on the tree and see the steps of decisionmaking process. Each node of the tree represents a mental space defined in the blending theory. The user can see the compression of blending theory by expanding the tree downward.

The user can also set the threshold values for speed, CPA, and the range evaluation process under a menu option. These values are used when digital evaluation is selected on the Analog & Digital Selection Panel. These values are used when digital evaluation is selected on the Analog & Digital Selection Panel.

The ADL Simulation can be used either in user mode or model mode. When user mode is selected, the identities of the tracks are changed by the user. We defined this mode for testing the simulation. This mode can also be extended to perform training of air-defense personnel. In this mode the competing models feedback panel does not show the weights of the competing models. When the model mode is selected the identities of the tracks in the simulation is changed by the model.



```
Regional Agent Events
    Regional Agent Coordianted Detachment Activities
    Regional Snooper Coordinated Attack Detachment Activities
    Regional Agent Merge Activities
        Time :143.0    Merge activity is detected between 0 and 1
```

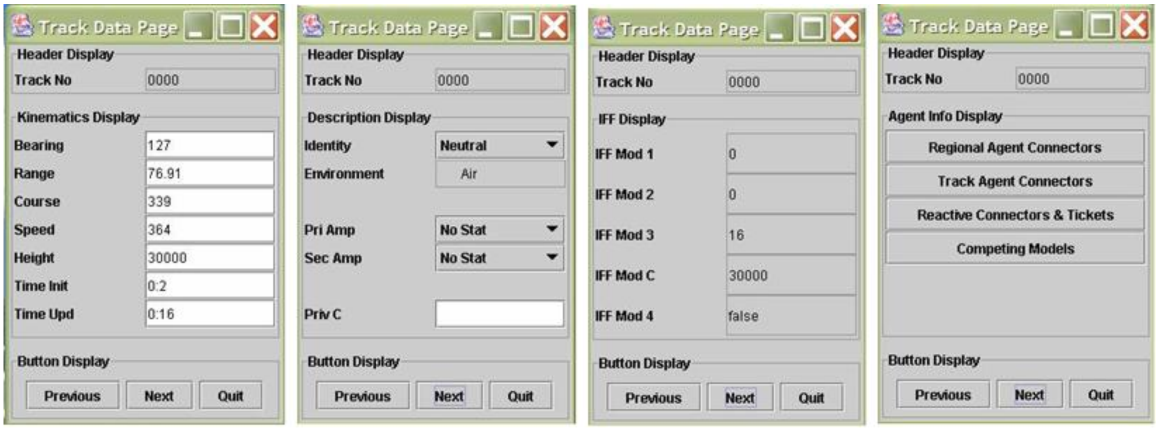
```
Track No :0 Knowledge Base
    Identity Node
        Friendly Ticket Node
        Civilian Ticket Node
        Unknown Ticket Node
        Suspect Ticket Node
        Hostile Ticket Node
        Time :5.7    Active model changed from Unknown to Hostile
    Split Node
        Time :36.1    Split Occured between track no 0 and track No 1
            Location information is received from track No :1
            My location is 309:455 and altitude is 5000
            Distance to other track is close
            Altitudes are close
            Tracks are not on an airbase
        Time :36.2    Split Occured between track no 0 and track No 1
    Detachment Node
    Snooper Node
    CPA Calculations
```

Figure 15. Evaluation of Model Panel

There are two submenus under the Intelligence menu: Setup IFF and Setup Threat Intelligence. IFF is used as an electronic identification method to identify air assets. IFF-1 values show the mission type of the aircraft. IFF-2 values are used to specify the aircraft identification number. IFF-3 values are used by both civilian and military aircraft to identify which Air Traffic Control Unit is controlling the aircraft. The user can choose to load default values. The user can define the threat expected sectors in the simulation via the Threat Intelligence Panel. This information plays a key role in the identification process.

D. TOOLBAR OPTIONS

A track can be selected by clicking on it. Once the track is selected a red square is drawn around it. The selected track's location, bearing, and range values can be observed on the Selected Track Information in the output panel. The "Track Data" button in the tool bar at the left side of the tactical display opens another track data frame.



Header Display
Track No 0000
Kinematics Display
Bearing 127
Range 76.91
Course 339
Speed 364
Height 30000
Time Init 0:2
Time Upd 0:16

Header Display
Track No 0000
Description Display
Identity Neutral
Environment Air
Pri Amp No Stat
Sec Amp No Stat
Priv C

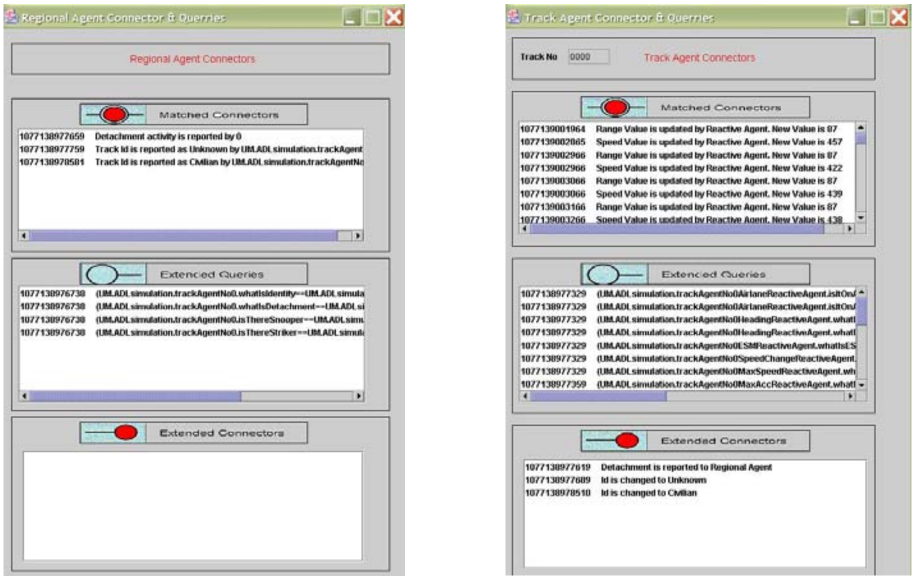
Header Display
Track No 0000
IFF Display
IFF Mod 1 0
IFF Mod 2 0
IFF Mod 3 16
IFF Mod C 30000
IFF Mod 4 false

Header Display
Track No 0000
Agent Info Display

Figure 16. Track Info pages

The track Info frame has four pages presenting sequentially kinematics, description, IFF, and agent info displays. The Description display is prepared for the user mode of the simulation. The user can change and add additional identity information of the track. The agent info page has access to four different frames: Regional agent connectors, track agent connectors, reactive agent connector and track agent tickets and finally competing models.

The regional agent connectors frame shows all the extended connectors, extended queries, and matched connectors of regional agents. The user can see the insight of the blending operation of the ADL Simulation at the regional agent level by traversing the data stored in this frame. Other frames show all the extended connectors extended queries and matched connectors of individual track agent and reactive agents. The user can see the current status of the identity tickets and ATO ticket on the menu selection of the reactive agent connectors panel.



```
1077138977659    Detachment activity is reported by 0
1077138977759    Track Id is reported as Unknown by UM.ADL.simulation.trackAgent
1077138978581    Track Id is reported as Civilian by UM.ADL.simulation.trackAgentNo
```

```
1077138976738    (UM.ADL.simulation.trackAgentNo0.whatIsIdentity==UM.ADL.simula
1077138976738    (UM.ADL.simulation.trackAgentNo0.whatIsDetachment==UM.ADL.si
1077138976738    (UM.ADL.simulation.trackAgentNo0.isThereSnooper==UM.ADL.simu
1077138976738    (UM.ADL.simulation.trackAgentNo0.isThereStriker==UM.ADL.simula
```

```
1077139001964    Range Value is updated by Reactive Agent. New Value is 87
1077139002065    Speed Value is updated by Reactive Agent. New Value is 457
1077139002966    Range Value is updated by Reactive Agent. New Value is 87
1077139002966    Speed Value is updated by Reactive Agent. New Value is 422
1077139003066    Range Value is updated by Reactive Agent. New Value is 87
1077139003066    Speed Value is updated by Reactive Agent. New Value is 439
1077139003166    Range Value is updated by Reactive Agent. New Value is 87
1077139003266    Speed Value is updated by Reactive Agent. New Value is 438
```

```
1077138977329    (UM.ADL.simulation.trackAgentNo0AirlaneReactiveAgent.isItOn/
1077138977329    (UM.ADL.simulation.trackAgentNo0AirlaneReactiveAgent.isItOn/
1077138977329    (UM.ADL.simulation.trackAgentNo0HeadingReactiveAgent.whatI
1077138977329    (UM.ADL.simulation.trackAgentNo0HeadingReactiveAgent.whatI
1077138977329    (UM.ADL.simulation.trackAgentNo0ESMReactiveAgent.whatIsES
1077138977329    (UM.ADL.simulation.trackAgentNo0SpeedChangeReactiveAgent.
1077138977329    (UM.ADL.simulation.trackAgentNo0MaxSpeedReactiveAgent.wh
1077138977359    (UM.ADL.simulation.trackAgentNo0MaxAccReactiveAgent.whatI
```

```
1077138977619    Detachment is reported to Regional Agent
1077138977689    Id is changed to Unknown
1077138978510    Id is changed to Civilian
```

Figure 17. Regional Agent and Track Agent Connectors Frames

The weight of each identity ticket can be observed on the Competing Models Frame. Each identity's weight is drawn by a figure with different colors. User can also zoom in and out during the simulation. There are four scales in the simulation.

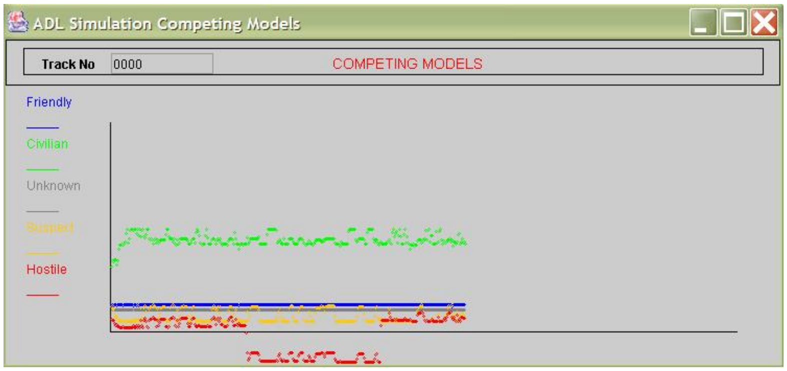
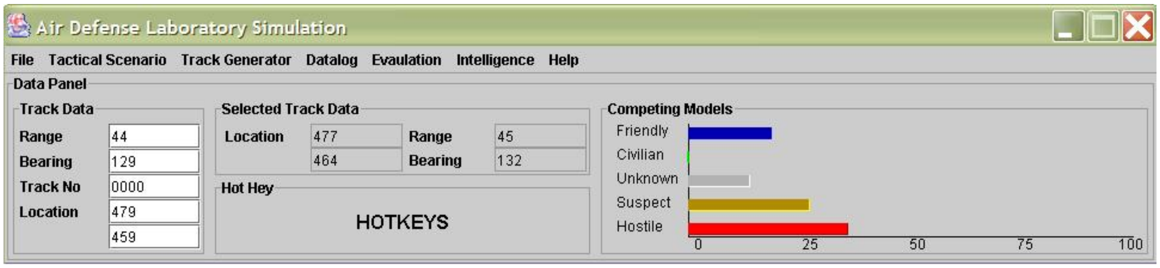


Figure 18. Competing Models Frame

The Simulation starts by creating the tracks based on the numbers entered in the Track generator panel. These tracks randomly pop up on the screen. The simulation may be paused by the pause button. Pause button puts all threads working in the simulator in sleep mode.

E. OUTPUT PANEL

The output panel has three subpanels: the data panel, the selected track panel, and the competing-models feedback panel. The data panel presents the location, range and bearing of the mouse on the tactical display. This panel also allows the user to enter a track number to set it as the selected track. After a track is selected by clicking on it or entering the track number into the Track Number field in the data panel, its location, bearing, and range values can be observed on the selected track subpanel. The competing-models feedback panel shows the current weight of all competing models.



Data Panel

Track Data

Range 44
Bearing 129
Track No 0000
Location 479
459

Selected Track Data

Location 477
464
Range 45
Bearing 132

Hot Hey

Figure 19. Simulation Output Panel

F. JAVA XML INTEGRATION

We integrated java and XML with JDOM beta9. JDOM is an open source API enabling writing, reading, and manipulating of XML files inside Java code. JDOM is not the only API available for Java-XML integration; there are DOM and SAX available for integration. We chose JDOM because it was simple to use and the JDOM API methods are following the same philosophy of existing Java API. It combines the advantages of two other API. JDOM is compatible with the classes of Java API. It is using the Collection class of Java API. The only drawback of JDOM is that JDOM is limited to Java. It is not a platform free API like DOM and SAX.

THIS PAGE INTENTIONALLY LEFT BLANK

V. DESIGN OF THE ADL SIMULATION PROGRAM

A. INTRODUCTION

There are two types of track agents in the ADL Simulation: Real track agents (modeling aircraft) and predictor track agents (modeling air-defense personnel). Real track agents have full control of the behaviors of the air tracks. They determine the actions that the aircraft will take. They have full access to track information and can modify them. Predictor track agents only can change the predicted identity value of the aircraft. They cannot change any other variable of the tracks. Predictor agents also can only receive the kinematics values of aircraft (location, speed, heading etc.), IFF values that the aircraft responds to interrogations, ESM values received from the aircraft, and intelligence information.

Predictor agents are designed in a layered structure to enable ease of implementation, to include every possible detail and to be able to use the membrane property of CMAS library. The membranes allow agents to operate in separate environments. There are three layers in the ADL Simulation: the regional-agent layer, the track-agent layer, and the reactive-agent layer. While there is one track agent and reactive agent for each track in the environment, there is only one regional agent for all tracks.

B. REAL TRACK AGENTS

Real track agents determine what actions that the aircraft will take at a specific location and time. Some of the decisions are determining the next waypoint, determining the point to turn, increasing or decreasing the speed and altitude, turning on/off the radar and IFF transponder, and commencing attack. There are seven main types of real track agents. The details of the implementation of the behaviors of the aircraft is based on our experience, tactical procedure publications, and air warfare game documents

1. Civilian Aircraft Track Agent

The civilian aircraft track agents' mission is to take off from an airport and follow air routes to the destination airport. Takeoff and destination airports are picked randomly. Typical values for a civilian aircraft are shown below.

Attribute	Behavior
Takeoff and destination airbases	Randomly chosen
Speed	Initial : 100 Max Speed: 400 + random number (1-75)
Max Acceleration	10-12
Altitude	Max 30000 Ft.
IFF-2	0
IFF-2	0
IFF-3	Random number (1-9999)
IFF-4	False
Radar Status	On
Radar Emission	Civilian Radar Emission
Turning angle	2°
IFF Transponder status	On

Table 3. Civilian Aircraft Attributes and Behaviors

A civilian track agent finds its waypoints by using a combination of A* search and Depth First Search (DFS). At each step, the track agent measures the distance to the next waypoint and determines the course to reach the next waypoint. When the aircraft comes to the turning point, it turns to a new waypoint with its turning angle. The turning angle is the amount the aircraft in one tenth of a second. Based on the speed and course, new location points are calculated and track position is set to these points. On the last waypoint the aircraft starts decreasing altitude and speed, and subsequently finishes its mission by landing at the destination airport.

2. Friendly Aircraft Track Agent

The friendly aircraft's mission is to fulfill a randomly chosen task. There are ten friendly tasks. Every friendly mission must be defined by an Air Tasking Order message. If no task is defined in such a message by the user, a randomly generated task is generated by the simulation and inserted in the message. The friendly tasks are as follows:

Task	**Behavior**
OCA Sweep	An Offensive Counter Air sweep mission is an Air-to-Air mission and its intent is to shoot down enemy aircraft.
RECON	A Reconnaissance mission’s intent is to take pictures in a given area or on a path.
CAS	A Close Air Support mission supports the army in ground attack missions.
BARCAP	A Barrier Combat Air Patrol mission’s intent is to protect a given area from enemy attacks.
Deep	A Deep mission’s intent is to attack enemy units in enemy territories.
OCA	An Offensive Counter Air mission’s intent is to attack enemy airfields.
BDA	A Battle Damage Assessment mission’s is to check the target status after the attack is completed.
SEAD	A Suppression of Enemy Air Defense mission’s intent is to attack enemy air-defense units such as SAM launchers.
DCA	A Defensive Counter Air mission’s intent is to protect AWACS or a High Value Unit (HVU) from enemy attacks.
ESCORT	An Escort mission’s intent is to protect a given unit from enemy attacks.

Table 4. Friendly Aircraft Missions [28]

The OCA Sweep and RECON missions are path missions. They have certain waypoints randomly chosen by the track agent. Other missions are area missions: Aircraft are given a certain area to stay inside during the task period. We did not implement the detailed specification of each task. From the point of view of an air-defense officer, an aircraft’s activity is less important once its identity is known. For that reason we divided the missions into path-based missions and area-based missions. The real track agent's mission is to make sure that aircraft are following the path or staying inside the area based on the mission type.

Attribute	Behavior
Mission	Randomly chosen
Speed	Initial : 100 Max Speed: 500 + random number (1-100)
Max Acceleration	10-17
Altitude	Based on mission. Around 20000 Ft.
IFF-1	Determined by ATO message
IFF-2	Determined by ATO message
IFF-3	Determined by ATO message
IFF-4	True
Radar Status	On
Radar Emission	Military Radar Emission
Turning angle	4°
IFF Transponder status	On
Origin	Safe sector

Table 5. Friendly Aircraft Attributes and Behaviors

Every friendly activity originates from a friendly country and their IFF-4 value is true with very high probability. When the mission is completed, the real track agent sends the aircraft toward the friendly country and it leaves the radar scope.

3. Enemy Aircraft Track Agent

An enemy aircraft track agent randomly picks one of the enemy missions and executes the requirements of this task. All of the enemy aircraft take off from the "threat expected" sector except for the aircraft that have chosen a terrorist attack mission. If there is no threat-expected sector defined in the simulation, the attack origin is selected randomly. These aircraft use randomly picked IFF-1, 2, and 3 settings. The IFF-4 value is false for enemy aircraft with very high probability. The enemy missions are:

Task	Behavior
HADB Attack	High Altitude Dive Bomb attack
DB Attack	Dive Bomb attack
Popup Attack	Low attack profile
Terrorist Attack	Specialized terrorist attack profile with a civilian aircraft

Table 6. Enemy Aircraft Missions [28]

An HADB attack is a medium attack profile. Aircraft begin to attack at 22,000 ft. and fly towards the ship. When the distance is 15 nm, the aircraft makes a 15° turn. When the ship is about 60-90° relative bearing from the ship's heading, the aircraft turns inbound and executes the attack on the ship. A DB attack is another medium altitude attack but it starts at an altitude of 10,000 ft. A pop-up attack is a low altitude attack. The aircraft approaches the ship at a low altitude and flies towards the ship until the range is 7 nm. At a 7 nm distance to the ship, the aircraft makes a 30° turn away from the ship. When the ship is at 30° relative bearing from the aircraft heading, the aircraft pulls up and increases its altitude. At 5000 ft. the aircraft turns inbound and starts decreasing its altitude and executes the attack. [28]

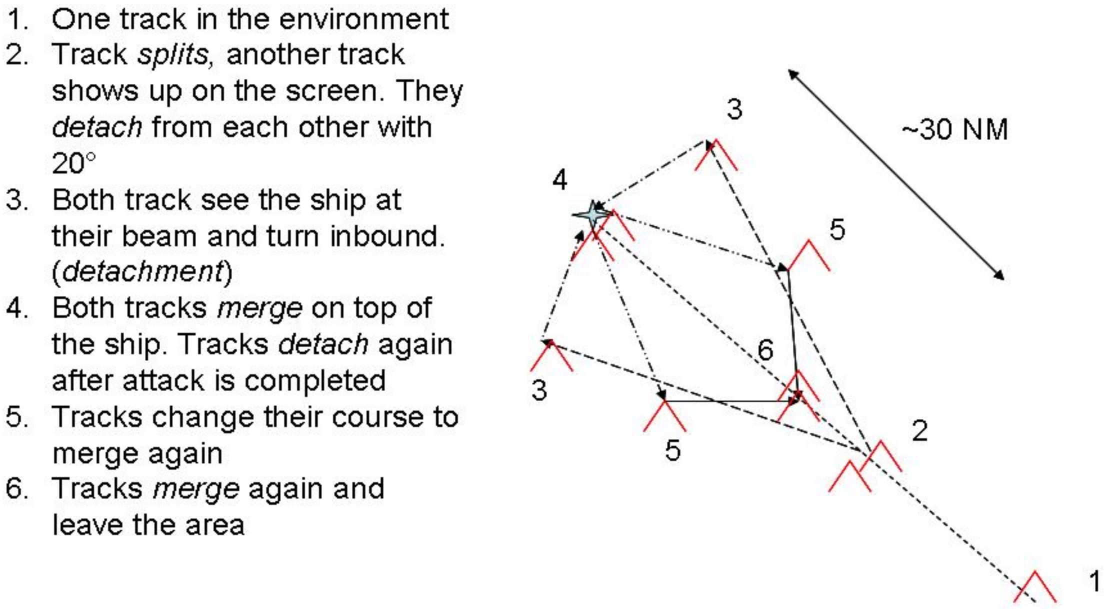
In a terrorist attack scenario, the enemy aircraft behaves like a civilian aircraft. It takes off from a randomly picked airport and its destination is another randomly picked airport. This scenario is prepared to suggest a terrorist attack like those of September 11, 2001. During its flight, the real track agent calculates the nearest point to the ship on the air route. At this point the aircraft suddenly turns inbound and decrease its altitude. This action gives the predictor agents a short reaction time to identify the aircraft's behavior. Terrorist attack specifications are otherwise the same as civilian aircraft. The enemy military aircraft specifications, except for terrorist attack missions, are as follows.

Attribute	Behavior
Mission	Randomly chosen
Speed	Initial : 100 Max Speed: 500 + random number (1-100)
Max Acceleration	10-17
Altitude	Based on mission. Around 20000 Ft.
IFF-1	Randomly chosen
IFF-2	Randomly chosen
IFF-3	Randomly chosen
IFF-4	False
Radar Status	On
Radar Emission	Military Radar Emission
Turning angle	4°
IFF Transponder status	On
Origin	Threat-expected sector

Table 7. Enemy Aircraft Attributes and Behaviors

4. Coordinated Detachment Attack Track Agent

The coordinated detachment attack track agent executes a coordinated attack on a ship. It was developed to test the coordinated activity detection of the regional agent. This agent handles three activities: detachment, split, and merges. At first only one aircraft is detected on the radar screen. Actually there are two aircraft but since they are so close to each other and the range is so long, the radar sense only one track. When the aircraft comes to about 30 nm, the track splits and another air track shows up on the screen. At this point, another enemy real track agent is created by the coordinated detachment attack track agent, but the control of the second enemy aircraft is under the first one. The first aircraft behaves as a wing commander in this scenario. When the aircraft split they detach from each other and fly 20° away from the previous course in different directions. The reason for this detachment is to prevent weapons coordination of the ship for self defense against the attack. When they are detached, the ship has to allocate different weapons and track radars for each of the aircraft. When the aircraft see the ship at their beam, they suddenly turn inbound and attack. This is called a coordinated attack. On top of the ship, the two aircraft merge and split again when they complete the attack. About 20 nm away from the ship these two aircraft merge once again and leave the area.



1. One track in the environment
2. Track *splits*, another track shows up on the screen. They *detach* from each other with 20°
3. Both track see the ship at their beam and turn inbound. (*detachment*)
4. Both tracks *merge* on top of the ship. Tracks *detach* again after attack is completed
5. Tracks change their course to merge again
6. Tracks *merge* again and leave the area

Figure 20. Coordinated Detachment Attack Profile

5. Missile Attack Track Agent

Air-to-Surface missiles (ASM) are the most dangerous airborne threat to a ship. They are fast, small, and difficult to detect and destroy. Additionally, they can be smart to avoid some of the Anti-Ship Missile Defense (ASMD) reactions of ship and they are high damage-capable. They can be released from an aircraft at about 40 nm away from the ship. This may make the ship's proactive reaction plan useless because they may not destroy the missile platform before the missiles are released because the platform is out of the effective weapon range of the ship.

The ADL Simulation has a special kind of agent to simulate the missile attack scenario. The missile attack track agent's mission is to lead the aircraft toward the ship and release its missile about 40 nm away from the ship. Then they turn away and leave the area. At the missile release point another track shows up on the radar screen, and is controlled by the missile track agent.

6. Missile Track Agent

The Missile track agent controls the missile track actions. We defined two types of missile profiles in the ADL Simulation. One of them is a sea-skimming attack missile and one is a pop-up attack missile profile. The sea-skimming missile flies just over the sea with an 80ft altitude. This prevents ship sensors from easily detecting and destroying it. The pop-up missile increases altitude suddenly at close range to the ship and then dives into ship. The reason is to drop explosive materials in the warhead of missile onto the ship to damage the ship's sensor and weapon systems even if the missile is shot by the ship's self-defense systems. Missile track agents randomly pick one of these two modes when the track is created.

7. User derived Track Agent

We provided a user-derived track capability to the ADL Simulation to add diversity of tracks. The user can define the track and behaviors on the Tactical Figure Control Panel. The user can select its location, speed, altitude, IFF values, and radar emission parameters. The changes in the behavior of the track are determined by geographical location and use waypoints. The user can define as many waypoints as he wants.

C. REACTIVE AGENTS

For each factor we defined as being important for air defense, we implemented a reactive agent to monitor its relevant data and inform predictor track agents of any changes in the data. We identified 17 reactive agents, all individual threads. These agents are created by track agents. The communication between track agents and reactive agents is provided by the CMAS library. Since we defined a different membrane for each track agent, the connectors extended by a reactive agent of a track cannot find a match with any other track agent besides the one by which they are created. Besides the majority of the factors defined in Liebhaber and Smith's research [29], we also defined our own factors as reactive agents in the ADL Simulation.

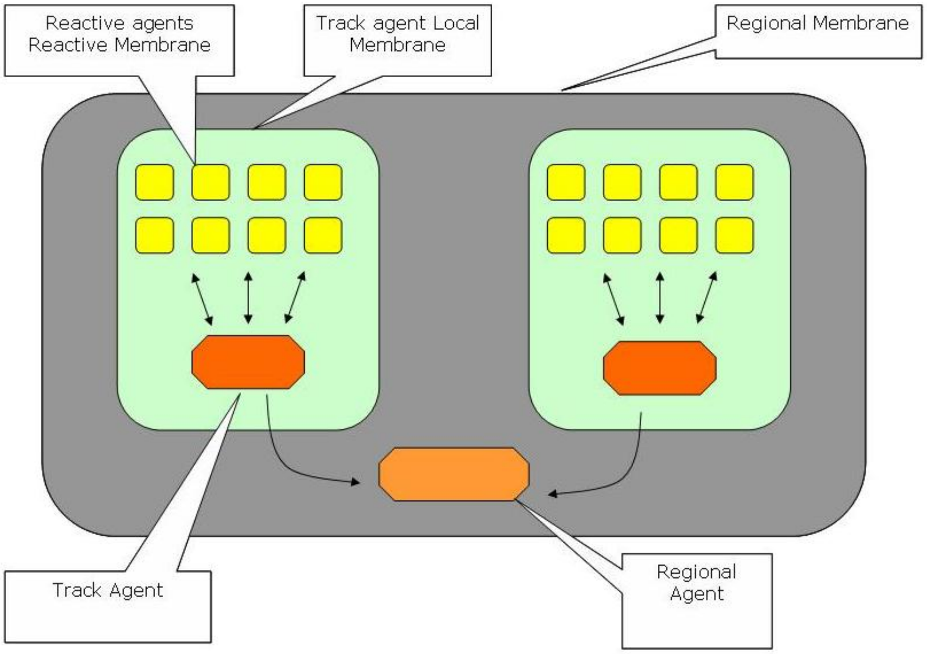


Figure 21. ADL Simulation Layered Structure

1. Airlane Reactive Agent

An Airlane reactive agent continuously compares the aircraft's location to air routes. Airlanes are standard commercial routes that civilian aircraft have to follow. Being on an airlane increases the probability that an aircraft will be civilian but it is not a guarantee. Airlane agents act differently under different threat levels. If the threat level is low, a reactive agent is more tolerant of an aircraft outside of an airlane.

The communication between airlane reactive agents and track agents is provided by the CMAS library connectors and queries. This reactive agent uses an Integer connector to send its information to the track agent. The track agent has a query for this connector: "Is the aircraft on the airlane?" The airlane reactive agent's connector matches with this query whenever reactive agent extends its connector.

Airlane reactive agents behave differently for analog and digital selection. In both selections, the reactive agent opens a window around the air track. The height and width of the window are determined by the threat level; a low threat level allows for a bigger window area. If digital is selected in the Analog & Digital Selection Panel, the Airlane reactive agent sets the connector to a 100 value when it finds a pixel in the search area occupied with an air route. Otherwise, the reactive agent sets the connector to zero. If analog is selected in the panel, the reactive agent looks for the nearest air route location in the search window. The reactive agent sets the connector with the value of the difference between the half of the diagonal distance of edges of window and the nearest range.

2. ESM Reactive Agent

Naval ships have an Electromagnetic Support Measurement equipment to detect electromagnetic emissions. Radio Frequency (RF) Radars have a fingerprint encoded in its frequency. If an ESM device detects this frequency and finds a match in its library, the ESM operator can identify the platform of the radar with its bearing. Civilian and military aircraft have specific navigation radars. Military fire-control radars and missile-seeker radars are working in higher frequencies than navigation and surveillance radars. In Liebhaber's research, experienced air-defense personnel used the ESM factor as the major identification factor in experiments [30]. If an aircraft turns on its fire-control radar, it is an obvious sign of a preparation for an attack on a ship.

Normally air-defense officers do not directly use the ESM equipment. An ESM operator uses this equipment, analyzes the data, and reports to the air-defense officer. The ADL Simulation works the same way. ESM reactive agents act like an ESM operator and report any changes to the track agent after analyzing the radar emissions received from the air track. We defined five different ESM reports in the simulation (see Table 5).

RADAR EMISSION TYPE	BEHAVIOR
Civilian aircraft radar emission	This is the typical civilian aircraft navigation radar. They usually work in low frequencies with respect to other kinds of radars.
Military aircraft radar emission	This is a typical military surveillance radar emission. The. ESM operator may identify the platform and possible threat to the ship based on this information.
Military aircraft fire-control radar emission	Fire-control radar works at the higher frequencies to detect and track the target. The target can identify a “lock on” operation with ESM equipment. Lock on is a precondition of an attack on a ship.
Missile seeker emission	Missile-seeker radars work in higher frequencies than fire-control radars because they need more precision to increase the probability of a “hit”. Missile-seeker detection by an ESM device is a sign of an attack on a ship.
No radar emission	An aircraft has turned off its radar.

Table 8. ESM Reactive-agent Messages to Track Agent

3. Heading-change Reactive Agent

The heading-change reactive agent reports the changes to the heading of the aircraft. Reactive agents use different levels of tolerance under different threat levels; under high threat levels they report small changes to track agents while under low threat conditions they do not. The reactive agent’s integer connector is extended when the heading change is more than the accepted limit based on the threat level status. In air defense, heading change becomes important when an aircraft suddenly turns inbound towards the ship to attack. In Liebhaber’s study, heading is the third most commonly used factor by air-defense personnel.

4. IFF Reactive Agents

IFF stands for Identification Friend or Foe. The IFF transponder devices are embedded in the aircraft and respond to the interrogations if they are on. IFF values are set before the flight and may be changed by the pilot during the flight. IFF categories and functions are listed in Table 9. In the ADL Simulation, we defined IFF-1, IFF-2, and IFF-3 reactive agents to track the IFF values of aircraft. These agents continuously interrogate the IFF values of the aircraft and report changes to track agents by extending their connectors.

IFF Mode	**Concept**
Mode I	IFF Mod I is used by military aircraft to show the mission of the aircraft.
Mod II	IFF Mod II is used to show the squadron of the aircraft that it belongs to. These numbers are unique and kept secret.
Mod III	IFF Mod III is used by both military and civilian aircraft. These values show which air traffic control is currently controlling the aircraft.
Mod IV	IFF Mod IV is an encrypted signal that can only be decrypted by a certain cipher. This cipher is kept secret and only friendly aircraft have the correct cipher.
Mod C	IFF Mod C shows the altimeter value of the aircraft. Not all the air surveillance radars are capable of three-dimensional signal processing. Therefore some of them can only locate the aircraft with ground reference systems.

Table 9. IFF System Modes and Concepts

The IFF-4 reactive agent continuously interrogates the aircraft and reports the results to track agents if it is different from the previous interrogation. When aircraft do not carry any IFF value to keep them undetected, IFF-4 becomes more important to recognize friendly aircraft. The IFF transponder status reactive agent checks the IFF transponder of the aircraft and reports this information to the track agent. Normally, all civilian aircraft keep their IFF transponders turned on. Military aircraft may turn off their transponders to keep them undetected.

5. Max Acceleration, Altitude, and Speed Reactive Agents

Military interceptors and fighter aircraft can reach higher accelerations than civilian aircraft. The max acceleration reactive agent calculates the maximum acceleration of the aircraft and reports this value to the track agent

The altitude reactive agent checks the altitude of the aircraft and informs the track agents of the changes in the altitude of the aircraft. Civilian aircraft usually navigate between 30,000-35,000 ft. on airlanes. Civilian aircraft have a maximum altitude limit in their specifications and cannot fly above these limits. Some military aircraft are designed to fly at high altitudes to avoid detection by air surveillance radars. Therefore altitude is an important factor in threat identification.

The max altitude reactive agent continuously checks the altitude and compares this value to current max altitude of the aircraft. If this value is changed, the reactive agent reports this change to the track agent with an Integer type connector. Some small aircraft and helicopters cannot fly over a certain altitude due to lack of sufficient air density.

The maximum speed of an aircraft is another factor used to predict the type of the aircraft. Military aircraft have larger maximum speed than civilian aircraft. For example Boeing-type commercial aircraft speeds vary between 0.75-0.9 mach (1 mach is 1067 km/h). A reactive agent continuously checks the aircraft speed and compares this value with its max speed value.

Commercial aircraft usually fly at 30,000 ft. altitude. They can cruise at lower altitudes only occasionally. However, for military fighters there is no limit for minimum altitude. A reactive agent checks the current altitude of the aircraft and reports this value if the current altitude is lower than the previously recorded minimum altitude.

The speed reactive agent checks the current speed of the aircraft and reports this value. The speed-change reactive agent continuously monitors the speed of the aircraft and reports significant speed changes. Its threshold values are determined by the threat level. Civilian aircraft usually maintain a specific speed during their cruise. Military aircraft speeds vary during the flight depending on the mission they are executing.

6. Origin Reactive Agent

The Origin reactive agent compares the origin of the aircraft with the intelligence information about the threat-expected sectors. In the ADL Simulation, with a very high probability the friendly aircraft originate from the friendly country sector while hostile aircraft originate from the threat-expected sectors. Threat intelligence can be set or observed under the Intelligence menu option.

Military aircraft carry a certain amount of load. This load includes fuel, weapons, and personnel. They can carry an extra tank for fuel to fulfill the requirements of the longer missions but then they cannot carry as many weapons. So military aircraft on strike missions carry a fuel amount as low as possible to fulfill the mission and carry as many weapons as possible. These aircraft take off, go directly to mission area, execute the mission, and return to the main base. Hence determining the origin of an aircraft is important, and all aircraft that have taken off from or detected in threat-expected sectors are always suspect.

7. Radar Status Reactive Agent

The radar status reactive agent follows radar emissions. If there is no radar transmission, the reactive agent extends its connector to the track agent and reports the radar status.

8 Random Number Finder Reactive Agent

This reactive agent generates a random number representing a numeric error, used to make the simulation more realistic. The accuracy of the data received about an air track decreases with range because of signal losses due to transmission impairments. These impairments include free-space loss, attenuation, attenuation distortion, fading, and multi-path propagations. The random number finder reactive agent determines a random number limit for all other reactive agents. This limit is determined by the range of the air track. Figure 22 shows the equation used to determine the error percentage limit in the ADL

Simulation: We used 30 for constant A. the following equations are our reasonable guesses for the evaluation of the relevant values.

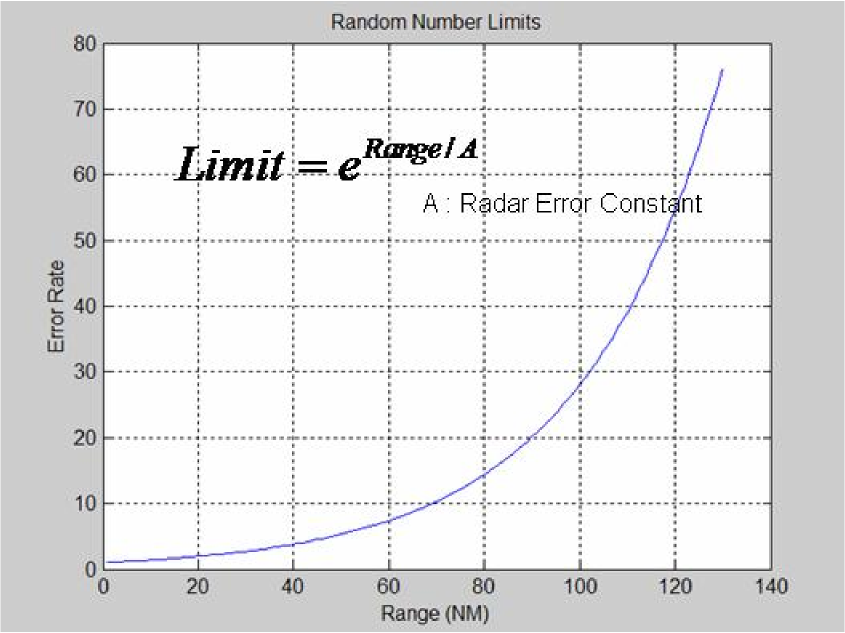


Figure 22. ADL Random Number Finder Reactive Agent Equation

For sample ranges the equation gives the following error percentage limits.

Range (nm)	**Random number limit**
128	71
64	8.44
32	2.90
16	1.7

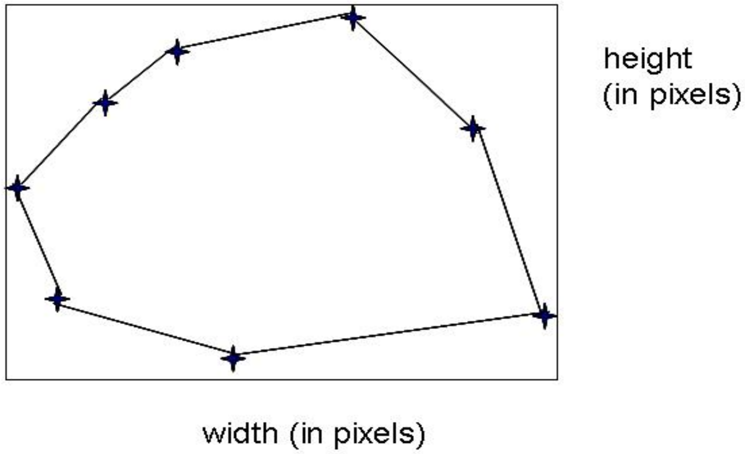
Table 10. Sample Error Percentage Limits

This means that at 128 nm range, the kinematics values that reactive agents receive from the air track are wrong 71% of the time in the ADL Simulation. Each time a reactive agent has to receive the kinematics values from the air track, they pick a random number between 0-100. If this number is more than the random number limit value determined by random number limit finder reactive agent, they get the kinematics value as it is. Otherwise they receive the value with an error. The error value is also determined randomly and added or subtracted from the actual value. Constant A in the equation may be changed for the type of the air radar. Since this thesis is unclassified, we could not use the error rate of actual air-surveillance radars. But the value that we used makes error rates that are reasonable.

9. Snooper Detector Reactive Agent

The snooper is a special aircraft type whose mission is to collect intelligence about the location of friendly aircraft and report this information to enemy headquarters. The snooper stays out of the ship's weapon range but most likely inside its sensory range. Snooper aircraft do not usually carry weapons to attack a ship. The presence of a snooper aircraft in the environment is a sign of a striker attack.

The snooper detector reactive agent is responsible for identifying snooper activities. The typical behavior of snoopers is to stay out of weapon range and orbit in a specific area. Reactive agents keep track of the reported locations in a two-dimensional array. We found the gradient magnitude of the locations of the aircraft to find the edges of the polygon that the aircraft is flying. We used four-neighbor centered formula to find the gradient magnitude. We then connect these edges to figure out the polygonal area that the aircraft is flying. If the density of aircraft locations within this polygon exceeds a threshold, we assume it fits into a typical behavior of a snooper (see Figure 23).



$$\text{Density} = \begin{cases} 2 \text{ x } (\text{height} + \text{width}) & \text{if } \dfrac{2 \text{ x } (\text{height} + \text{width})}{\text{Number of pixels that aircraft flied over}} > 1 \\ 0 & \text{otherwise} \end{cases}$$

Figure 23. Snooper Detector Reactive Agent Equation

D. PREDICTOR TRACK AGENTS

For each track in the environment one predictor track agent is created. The mission of track agents is to predict the identity and the potential intention of the aircraft. They have limited access to track data, to only the data that an air-defense team in the CIC receives from the air track and intelligence. Predictor track agents can retrieve the kinematics of air tracks including location, speed, heading, altitude, IFF values and ESM detections.

Predictor agents are located in the middle level of three-layer structure of the ADL Simulation. They receive the information from reactive agents and blend all this information to predict the identity of the aircraft. Predicted identity, detachment, detected snooper behavior, hostile activity, and location information are reported to regional agents via connectors.

1. Predictor Agent Connectors and Queries

Predictor track agents use the CMAS library to communicate with both reactive agents and regional agents. The queries and connectors are as in Table 11.

Type	Value	Corresponding Match	Criteria
Identity Connector CMAS Library Integer Value	Predicted identity of the aircraft: 1 Friend 2. Civilian 3 Suspect 4 Hostile 5 Unknown	Regional agent “What is identity” query	Connector is extended only when the predicted identity is changed
Snooper Connector CMAS Library Boolean Value	True/ False	Regional agent “Is there Snooper” query	Predictor agent extends this connector if snooper detector reactive agent extends its connector
Striker Connector CMAS Library Boolean Value	True/False	Regional agent “Is there striker” query	Predictor agent extends this connector if predicted identity is hostile
Detachment Connector CMAS Library Integer Value	A protocol between the predictor agent and the regional agent to transfer detachment information. Value=1000000 x track No+ 1000 x Location X + Location Y Value=****-***-*** * : Track No * : X Location information * : Y Location Information	Regional agent “Is there detachment:” query	Predictor agent extends this connector when a heading change is reported by heading-change reactive agent
Location Connector CMAS Library Integer Value	Location information is transferred to other predictor agent. This value is used to recognize a split operation at the regional-agent level.	Predictor agent “ what is location” query	Predictor agent extends this connector once when it is created.
Threat Level Connector CMAS Library Integer Value	Numeric threat level information. Threat level is received from regional agent.	Reactive agents “what is threat level query”	When threat level is updated by regional agent
“Is it on airlane” Query	The closeness to the nearest air route	Airlane reactive agent airlane Integer type connector	Airlane reactive agent extends its connector when the aircraft is on airlane
“What is heading change” query	Heading change of air track	Heading change reactive heading-change agent Integer type connector	Heading-change reactive agent extends its connector when the heading change is more than a threshold value
“What is heading” query	Heading value of the air track	Heading-change reactive agent Integer type heading connector	Heading-change reactive agent extends its connector each time heading is updated
“What is ESM” query	Radar emission received from the air track	ESM reactive agent Integer type ESM connector	ESM reactive agent extends its connector when the radar emission received from the air track is changed

“What is speed change” query	Change at speed of the air track	Speed change reactive agent speed change Integer type connector	Speed change reactive agent extends is connector when the change at speed exceeds a threshold value
“What is max speed” query	Max speed of the air track	Speed reactive agent max speed Integer type connector	Speed reactive agent extends its connector when the speed of the air track is greater than current max speed
“What is max acceleration” query	Max acceleration of the air track	Max Acceleration reactive agent max acc Integer type connector	Max Acc reactive agent extends its connector when the calculated max acc is greater than current max acc
“What is speed” query	Speed of the air track	Speed reactive agent speed Integer type connector	Speed reactive agent extends its connector when the speed of the aircraft changes
“What is IFF-1” query	IFF-1 value of the air track	IFF-1 reactive agent IFF-1 Integer type connector	IFF-1 reactive agent extends its connector when the IFF-1 value received from the air track is changed
“What is IFF-2” query	IFF-2 value of the air track	IFF-2 reactive agent IFF-2 Integer type connector	IFF-2 reactive agent extends its connector when the IFF-2 value received from the air track is changed
“What is IFF-3” query	IFF-3 value of the air track	IFF-3 reactive agent IFF-3 Integer type connector	IFF-3 reactive agent extends its connector when the IFF-3 value received from the air track is changed
“What is IFF-4” query	IFF-4 value of the air track (True/False)	IFF-4 reactive agent IFF-4 Boolean type connector	IFF-4 reactive agent extends its connector when the IFF-4 value received from the air track is changed
“What is IFF Transponder status” query	IFF transponder status of the air track	IFF Transponder status reactive agent IFF transponder status Boolean type connector	IFF transponder status reactive agent extends when the status of transponder is changed
“What is Radar Status” query	Radar status of the air track	Radar status reactive agent radar status Boolean type connector	Radar status reactive agent extends its connector when the radar status is changed
“What is altitude” query	Altitude of the air track	Altitude reactive agent altitude Integer type connector	Altitude reactive agent extends its connector when the altitude of the air track is changed
“What is max altitude” query	Max altitude of the air track	Max altitude reactive agent max altitude Integer type connector	Max altitude reactive agent extends its connector when the altitude of the air track is greater than current max altitude value
“What is min altitude” query	Min altitude of the air track	Min altitude reactive agent min altitude Integer type connector	Min altitude reactive agent extends its connector when the altitude of the air track is less than current min altitude value
“What is origin” query	Origin of the air track: 0 Northeast 1 North 2 Northwest	Origin reactive agent origin Integer type connector	Origin reactive agent extends its connector once when the predictor agent is created

	3 West 4 Southwest 5 South 6 Southeast 7 East		
“Is it snooper” query	True/False	Snooper reactive agent “it is snooper” Boolean type connector	Snooper reactive agent extends its connector when it decides that the behavior of air track fits into snooper behaviors
“What is threat level” query	Threat level of air warfare: 0 White 1 Yellow 2 Red	Regional agent “threat level” Integer type connector	Regional agent extends its connector when it changes the threat level of air warfare
“What is your position” query	Position information of other tracks	Predictor agents “Location” Integer connector	Predictor agent extends its connector once when it is created.
“Is there merge” query	Track numbers of the air tracks involved in the merge operation	Regional agent “merge occurred” Integer type connector	Regional agent extends its connector when a merge operation is detected
“Is there coordinated detachment” query	Track numbers of the air tracks involved in the coordinated detachment operation	Regional agent “coordinated detachment occurred” Integer type connector	Regional agent extends its connector when a coordinated detachment operation is detected

Table 11. Predictor Track Agent Connectors and Queries

2. Predictor Agent Competing Models

There are five competing models inside each predictor track agent: Civilian, Unknown, Friendly, Suspect and Hostile. These models are created at the beginning of the simulation for each track as a ticket. At each cycle the predictor agent calculates the weight of these tickets. The model with highest weight for a track is the active model and predicted identity. Model weights for a specific track can be observed on the feedback panel in our implementation by selecting the air track. The Unknown identity is default active model with 0.5 weight. All other models start with 0.0001 weight.

3. Predictor Agent Tickets

A predictor agent has two main kinds of tickets: identity and independent. We defined a ticket for each aircraft identity in the ADL Simulation. Independent tickets are not related directly to identity tickets but the procedures they execute affect the data values of the frames of the identity tickets.

A civilian identity ticket contains six data frames: the ESM frame, the altitude frame, the speed frame, the airlane frame, the IFF evaluation frame, and the origin frame. These frames except the IFF evaluation frame are set when a match occurs between queries and corresponding reactive agent connectors. The IFF evaluation frame is set by the IFF Evaluation independent procedural ticket.

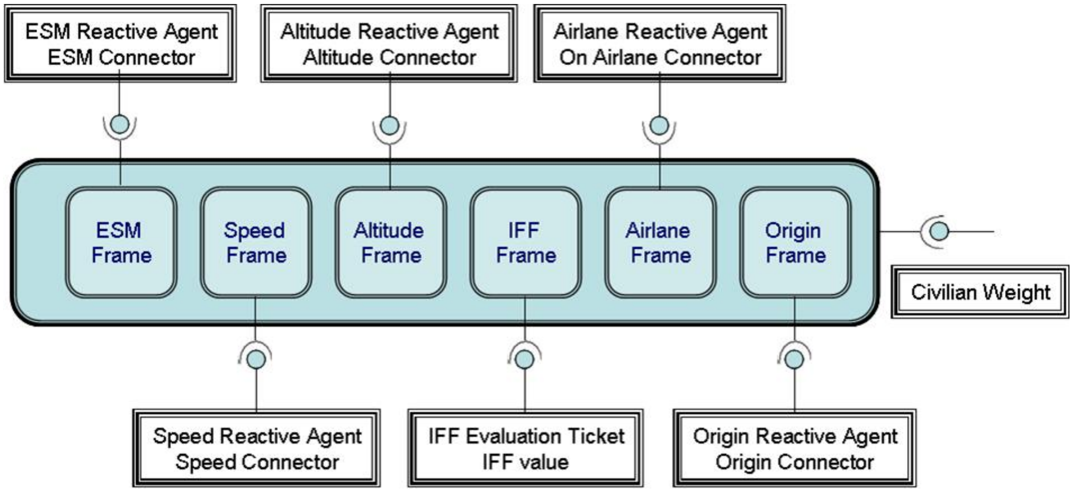


Figure 24. Civilian Ticket and Frames

A Friendly identity ticket has four data frames: the ESM frame, the IFF evaluation frame, the origin frame, and the ATO frame. The ESM and origin frames are set by a match with corresponding reactive agent; the IFF evaluation and ATO frames are set by independent IFF evaluation and ATO evaluation tickets.

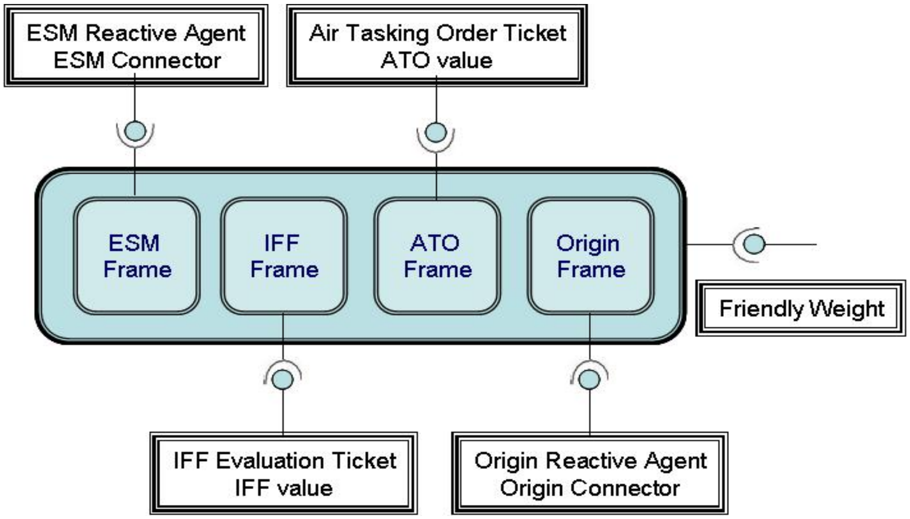


Figure 25. A Friendly Ticket and Frames

The Hostile identity ticket has ten data frames: the ESM frame, the range frame, the altitude frame, the airlane frame, the CPA frame, the origin frame, the IFF evaluation frame, the speed frame, the max speed frame, and a combination of the altitude, range and CPA frames. All frames except the IFF evaluation, CPA, and combination frames are set by a match with corresponding reactive agent connectors. Others are set by independent tickets.

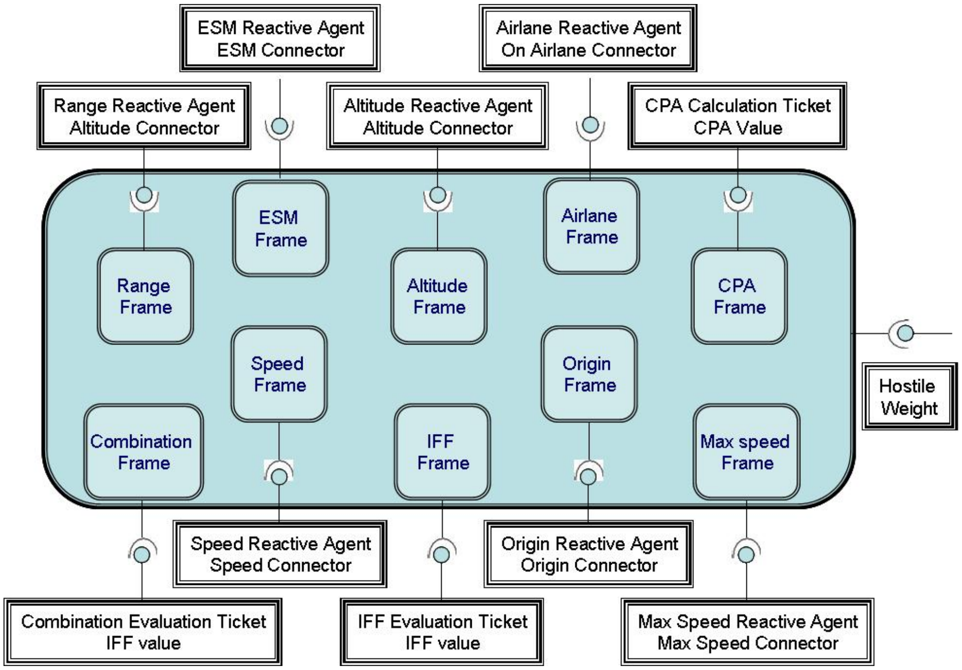


Figure 26. Hostile Ticket and Frames

A Suspect identity ticket has five data frames: the IFF evaluation frame, the altitude frame, the origin frame, the speed frame, and the max speed frame. All frames except the IFF evaluation frame are set by corresponding reactive agent connectors.

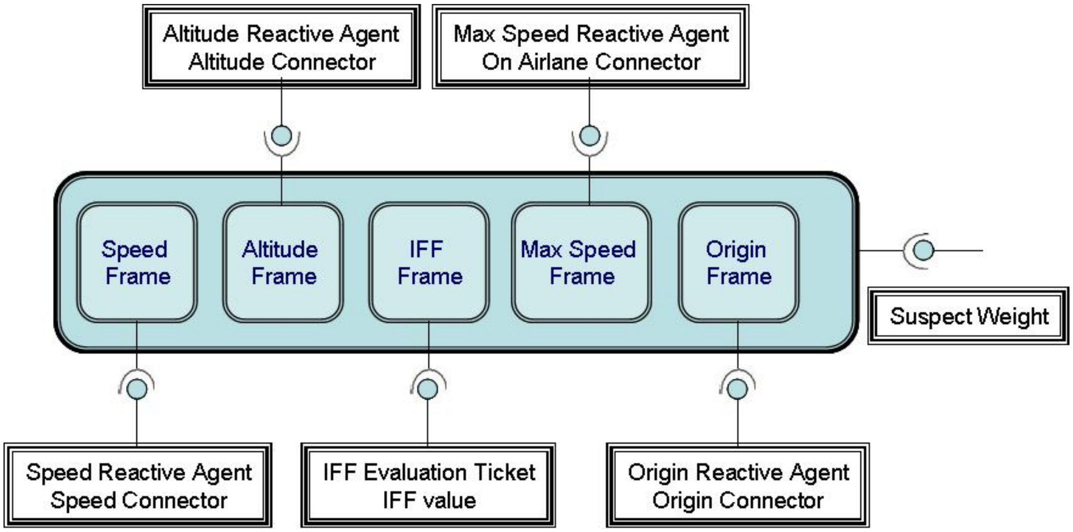


Figure 27. Suspect Ticket and Frames

An Unknown identity ticket has two data frames: the IFF evaluation frame and the ESM frame. The ESM frame is set by the ESM reactive agent while the IFF Evaluation frame is set by the IFF Evaluation independent ticket.

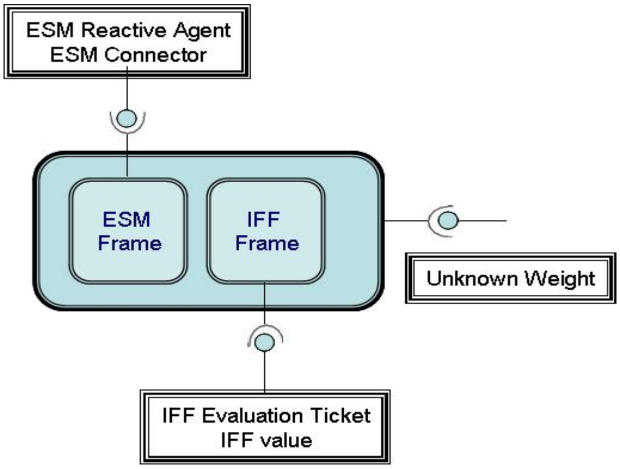


Figure 28. Unknown Ticket and Frames

The ADL Simulation has also four independent tickets. Once these tickets are completed they are blended anytime one of their frame data value is updated. These tickets are completed when all its data frames are set. The independent tickets are the IFF evaluation ticket, the CPA calculator ticket, the ATO ticket, and the combination ticket for hostile identity. These tickets create their own local integration networks and eventually connect with the identity tickets integration network.

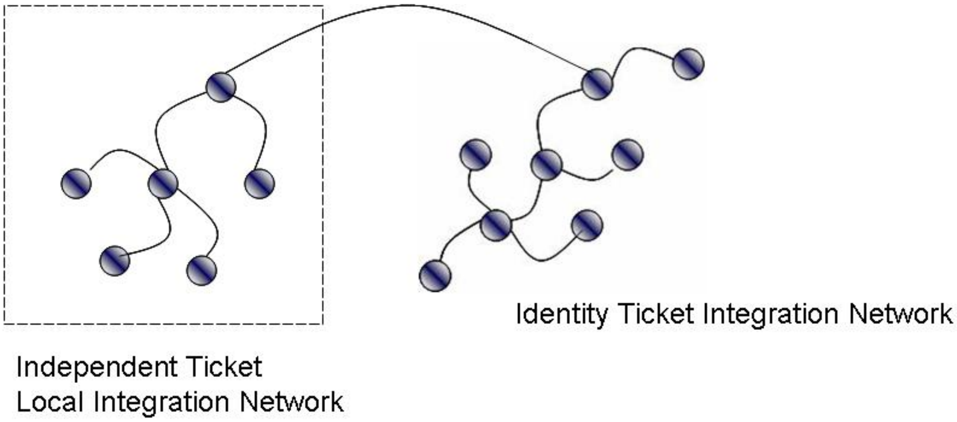

Identity Ticket Integration Network
Independent Ticket
Local Integration Network

Figure 29. Connecting Local Independent Ticket Integration Network to Identity Integration Network

An IFF Evaluation ticket has four data frames: IFF-1, IFF-2, IFF-3, and IFF-4. These frames are set by reactive agent connectors. An IFF Evaluation ticket calculates a weight for each identity ticket based on the values of its data frames.

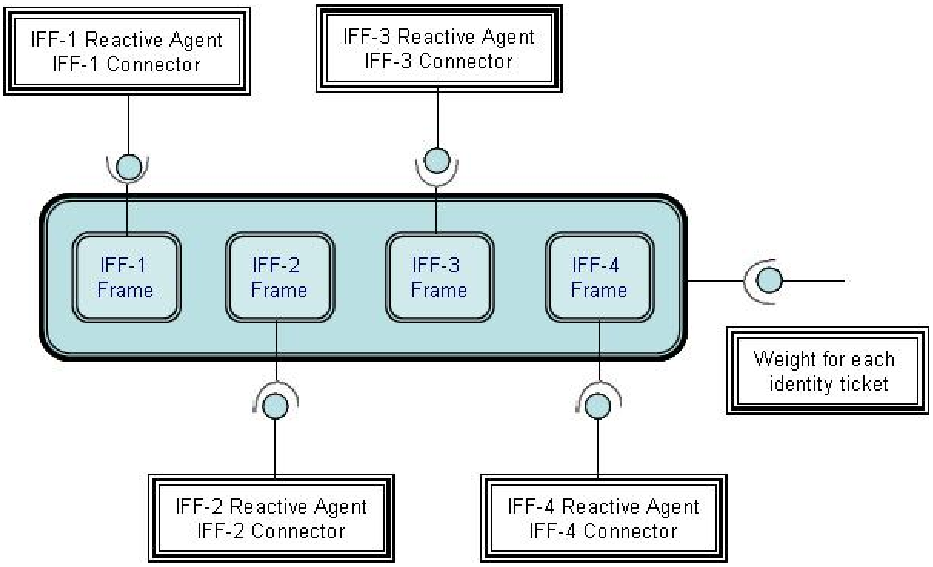


Figure 30. IFF Evaluation Independent Ticket

The CPA calculator ticket has only one frame: heading change. Anytime a heading-change reactive agent extends its connector, the CPA calculator independent ticket calculates a new CPA value based on the new heading value of the air track. The ATO Evaluation ticket has six frames: IFF-1, IFF-2, IFF-3, heading, location, and time frames. These frames are set by corresponding reactive agent connectors. When one of the frames is set, the ticket is executed and the result sets the ATO Evaluation frame of identity tickets. The Combination independent ticket has three data frames: Altitude, CPA, and range frames. The CPA frame is set by CPA calculator independent ticket and others are set by corresponding reactive agents. If aircraft is inbound, its range is close, and its altitude is low, a combination ticket sets the combination frame of the hostile ticket to true. That adds extra weight to the hostile identity ticket.

4. Weighting Procedure for the Predictor Track Agent

The Predictor track agent calculates the weights of each identity ticket. The ticket with the highest weight becomes the active model and predicted identity of the track.

a. Weighting the Civilian Ticket

The expected ESM behavior from a civilian aircraft is either a civilian navigation radar emission or no radar emission. If one of these values is received from ESM reactive agent the weight is increased, otherwise decreased.

The affect of the air track's altitude is different for digital and analog selection of the simulation. For the digital evaluation, the threshold values are 25,000 and 35,000 ft. since civilian aircraft usually fly between these levels. Therefore if the altitude of the air track is between these levels, the weight of the civilian ticket is increased. If analog evaluation is selected, the formula 5*exp(Altitude*0.0001) is used to calculate the altitude addition to weight of the ticket for altitude values less than 35,000 ft.

If the aircraft is on airlane the weight is increased if digital evaluation is selected. If analog evaluation is selected, airlane reactive agent sets the connector with a value proportionate to the range of the nearest air route point to location of the air track. If aircraft is not on airlane, the weight is decreased because it is a requirement to follow air routes.

The typical speed value for civilian aircraft is between 0.76-0.89 mach. If digital evaluation is selected, the acceptable spectrum for civilian aircraft speed is between 400 and 500 knots. If analog is selected for evaluation, the formula 100*sin((Speed-400)*1.81) is used to find the value to add the weight of the ticket for the speed values between 400 and 550 knots.

If the aircraft took off from a place not in the threat-expected sector, the weight of the ticket is increased. The IFF evaluation independent ticket evaluates the current IFF values and finds a weight for civilian ticket. All these weights are added up and normalized to find to total weight of the ticket.

b. Weighting the Friendly Ticket

ESM devices carry radar fingerprints in their libraries. If the ESM operator finds a match with a known fingerprint, he can tell even the name of the platform. In the simulation we assumed that our ESM library is not complete yet.

Therefore in a friendly ticket, the ESM frame only distinguishes military and civilian aircraft, and among the military aircraft it distinguishes the ones with hostile intention.

In the simulation, we assumed that all the friendly aircraft take off from a place with a very high probably that a threat is not expected. The threat-expected sector is defined before the simulation is started. If an aircraft takes off from a safe place, the friendly ticket's weight is increased, otherwise decreased. The IFF evaluation ticket evaluates a weight for friendly ticket for the current IFF values. The ATO evaluation ticket checks the aircraft behaviors with all friendly missions defined in the ATO. If there is a mission in ATO that matches with the actions of the aircraft, the weight is increased. The total weight is then normalized.

c. Weighting the Hostile Ticket

Air-defense personnel identify most of the threats against ships by using ESM devices. Missile-seeker radar, a fire-control radar locked on the ship, or military surveillance radar searching in the area are all signs of a threat for the ship. For that reason, the ESM frame in hostile ticket has more effect on the weight of the ticket than other frames.

Air defense of a ship goes from the highest priority threat to lowest priority one. Highest priority threats are the ones that show an immediate threat against ship. They are usually the ones closest to ship. The range frame evaluates the range and increase the weight of the ticket based on the range of the aircraft. If digital evaluation is selected, the range is compared to a threshold value. If range is less than that threshold, the weight is increased. The threshold value can be set on the range threshold setting panel under Evaluation menu option. If analog evaluation is selected the formula 100*exp(-0.015*Range) is used to find the range effect on weight of the ticket.

The Altitude frame is also evaluated differently based on the selection of analog or digital. If digital evaluation is selected, we defined three threshold values of 10000, 20000, and 30000 ft. The ticket adds a different weight based on the altitude value of the air track. If analog evaluation is selected the formula 150*exp(-Altitude*0.0001) is used to find the weight.

The Airlane frame is set by airlane reactive agent. The weight of airlane frame is the inverse of the value that the reactive agents set the frame. The closer to airlane is the less weight for airlane frame of hostile ticket.

For the CPA distance frame, if digital evaluation is selected, a threshold value is used to determine the weight of the frame. This threshold value can be set on the CPA threshold selection panel under the Evaluation menu. If analog evaluation is selected the formula 150*exp(0.02*CPA) is used to calculate the weight of the frame.

The speed frame of the ticket is another one evaluated based on the selection of analog or digital approach. If digital evaluation is selected, speed is checked against a threshold value. This threshold value can be set on the Speed Threshold Selection Panel dropdown menu under the Evaluation menu option. If speed is greater than threshold the weight of the ticket is increased. If analog evaluation is selected, the formula 15*exp(0.02*(speed-450)) is used to calculate the weight of the frame.

Most hostile activities originate from the threat-expected sector. Therefore the weight of hostile ticket is increased for air contacts originating from a hostile direction. The IFF Evaluation ticket calculates a weight for the hostile ticket based on the current IFF values of the air track. We also used a combination frame (subframe) of three frames of the hostile ticket. If air track is inbound at low altitude at close range, the weight of the air track is increased. These three frames behave like an internal ticket inside hostile ticket. If the max speed of the air track is more than expected max value from a civilian aircraft the weight of ticket is also increased.

d. Weighting the Suspect Ticket

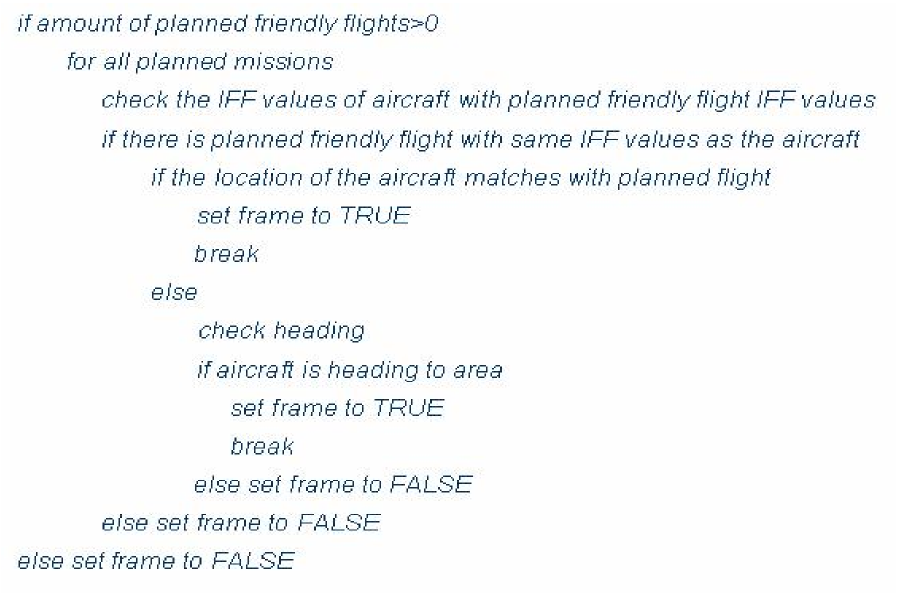
We used the same algorithm for evaluating the altitude, speed and maximum speed frames of the suspect ticket as the hostile ticket frames for both digital and analog evaluations selections. Suspect identity is a first step of hostile identification. Air-defense officers usually first identify an air track as suspect if there is not much hostile activity evidence. As the hostility evidences increases, then they identify the track as hostile.

e. Weighting the Unknown Ticket

There are only two frames in the unknown identity ticket. The Unknown identity is default identity for any emergent track on the radar screen. This means that there is not much evidence to identify the air track as one of the other four identities. Air-defense officers do not tend to leave this identity on air track for long and they try to change it as soon as they can. For that reason we defined only two frames for this ticket, the ESM frame and the IFF evaluation frame.

f. Execution of the ATO Ticket

The ATO ticket sets the ATO frame of the friendly identity ticket. After the ticket is completed the following pseudocode is executed in the ticket:



```
if amount of planned friendly flights>0
    for all planned missions
        check the IFF values of aircraft with planned friendly flight IFF values
        if there is planned friendly flight with same IFF values as the aircraft
            if the location of the aircraft matches with planned flight
                set frame to TRUE
                break
            else
                check heading
                if aircraft is heading to area
                    set frame to TRUE
                    break
                else set frame to FALSE
        else set frame to FALSE
else set frame to FALSE
```

Figure 31. Evaluation of ATO Frame of Hostile Ticket

g. *Execution of IFF Evaluation Ticket*

The IFF Evaluation ticket has four IFF frames. The execution of the ticket means finding the meaning of combination of four IFF frames. IFF Mod I and IFF Mod II may be set or not set. If they are set they may be right or wrong. That makes total of three possibilities for each IFF Mod I and IFF Mod II. IFF Mod III may be set or not set; hence there are two possibilities for IFF Mod III. IFF Mod IV may be right or wrong. These conditions create 36 different combinations for IFF Evaluation ticket.

$$\frac{IFFModI}{3} \times \frac{IFFModII}{3} \times \frac{IFFModIII}{2} \times \frac{IFFModIV}{2} = \frac{1}{36}$$

We defined a table for all these possible combinations. This table includes a weight value for each identity ticket for each combination. IFF Frames of identity tickets retrieve these weights. The air-defense team on a ship does the same procedure for IFF checking: Once the aircraft responds to IFF interrogations, the air-defense team checks the received IFF values with the values in published books or ATO message.

h. Execution of CPA Ticket

The CPA ticket calculates the CPA distance of an air track. The CPA is the closest point that the air track will pass by the ship. The ADL Simulation's inbound decision about an air track depends on the CPA distance. The CPA ticket first finds the real bearing of the ship from the air track. The ticket then finds the difference between this bearing and heading value of air track. The heading value is provided by heading reactive agent heading connector. If the difference is 90° the current point is the closest point. If it is more than 90°, the closest point had already been passed otherwise the tangent of the difference is the CPA distance of the air track. This value is provided to CPA frames of identity tickets.

i. Split Activity Detection

The split activity detector ticket has two location frames. The first is set by another predictor track agent location connector. This connector is set once by the other predictor track agent when the other track is first created. The other location frame is set by track location data and the ticket is executed. If the other track location is found close to the first track location, a split connector is extended. The track numbers of tracks involving into split operation is then broadcast to all predictor track agents via split connector.

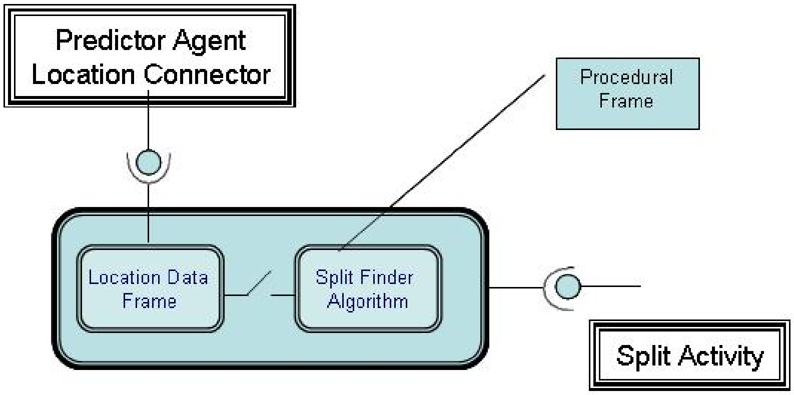


Figure 32. Predictor Agent Split Activity Detector Ticket

E. THE REGIONAL AGENT

The regional agent works at top of the layered structure of the ADL Simulation. There is only one regional agent in the simulation. Its mission is to find coordinated activities between tracks and regional activities involving more than one track. We defined three regional agent activities in the simulation: Snooper supported attack activity, coordinated detachment activity, and merge activities. In a snooper supported coordinated activity, at first snooper appears in the environment. It is believed that snooper reports the ship location to air force units and then enemy air force strikers comes into the environment to attack the ship. This is a coordinated activity between snooper and a striker. In a coordinated detachment activity, there are two enemy aircraft involved. Both of them act in coordination when they are turning. Merge activity is the joining of the two tracks. Based on these activities, the regional agent determines the threat level and broadcasts it to all track agents via threat level connector. The connectors and queries of regional agent are as follows:

Type	Value	Corresponding Match	Criteria
Threat-level Connector CMAS Library Integer Value	Threat level of the environment: 1 White 2 Yellow 3 Red	Predictor track agent "What is threat level" query	Connector is extended when regional agent changes the threat level
Coordinated detachment Connector CMAS Library Integer Value	A protocol between regional agent and predictor agent to transfer coordinated detachment information. Value=1000 x track No 1+ track No 2 Value=****-**** * : Track No 1 * : Track No 2	Predictor agent "Is there coordinated detachment" query	Regional agent extends the connector when there are two different detachment activities at close ranges and near-simultaneous times
Merge Connector CMAS Library Integer Value	A protocol between regional agent and predictor agent to transfer coordinated detachment information. Value=1000 x track No 1+ track No 2 Value=****-**** * : Track No 1 * : Track No 2	Predictor agent "Is merge" query	Regional agent extends the connector when there are two tracks at close location and altitude
"What is identity" query	Identity of the air track	Predictor track agent Identity Integer type Connector	Predictor agent extends the connector when the predicted identity changes

“Is there snooper” query	True/False	Predictor track agent snooper connector	Predictor track agent extends the connector when a snooper typical behavior detected
“Is there striker” query	True/False	Predictor track agent striker connector	Predictor track agent extends connector when predicted identity is hostile
“Is there detachment” connector	Detachment information of the air track. Value=1000000 x track No 1+ 1000 x Location X + Location Y Value=****-****-**** * : Track No * : Location X * : Location Y	Predictor agent detachment connector	Predictor track agent extends the connector when the track changes its heading
“What is location” query	Location of the air track Value=1000000 x track No 1+ 1000 x Location X + Location Y Value=****-****-**** * : Track No * : Location X * : Location Y	Predictor agent my location connector	Predictor agent periodically extends its location connector

Table 12. Regional Agent Connectors and Queries

The regional agent has three tickets: the snooper detector ticket, the merge detector ticket, and the coordinated detachment detector ticket. The snooper detector ticket has two frames: a snooper frame and a striker frame set by predictor agent snooper and striker connectors. The merge detector ticket has two location frames set by my location connectors of predictor agent. The coordinated detachment ticket has two detachment frames.

The snooper detector ticket is completed by setting both of its frames. The snooper frame should be set before the striker frame is set. Therefore this is a synchronous ticket. Once the snooper frame is set, a striker query is extended by the ticket. The ticket extends its snooper coordinated activity connector when the striker frame is set after snooper is set. This is a typical engagement with a third party unit where the snooper plays the role of a target report unit. After a snooper is detected in the area by one of the predictor track agents, the threat level is increased to yellow if it is white since the existence of a snooper in the area is the sign of upcoming strikes. This is called generalization in blending theory.

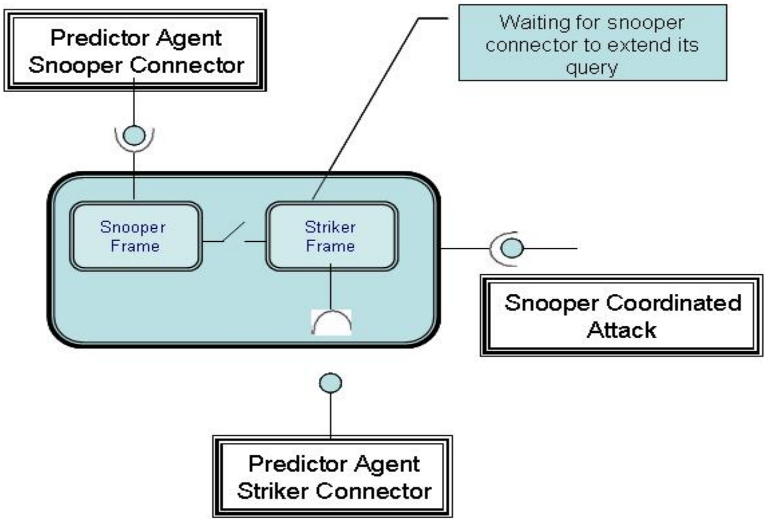

Predictor Agent Snooper Connector
Waiting for snooper connector to extend its query
Snooper Frame
Striker Frame
Snooper Coordinated Attack
Predictor Agent Striker Connector

Figure 33. Regional Agent Snooper Detector Ticket

The merge detector ticket is executed each time location information is received from one of the predictor agents. One of the frames is set by this match if this is not the first location report to the regional agent because a track reported for the first time cannot merge with another track. Other frames are set by other track locations in sequence. The ticket is then executed for each other track location. The merge connector is extended if the ticket finds another track location with a close distance and similar altitude to first frame location and altitude. The first frame is a data frame holding the location data of the reporting predictor agent's track, and other frame is a procedural frame executing the ticket for all other track locations

The coordinated detachment detector ticket finds two detachment activities reported by predictor agent that are close ranges to ship at close times. Each time a predictor agent reports a detachment activity, this activity is stored in a data structure and ticket compares the reported detachment to all detachment activities in this data structure. The data structure is a stack and data traverse starts from the last imported data. When ticket cannot find a match within an acceptable time threshold value, it stops traversing the stack.

F. BLENDING THEORY AND THE ADL SIMULATION

As we discussed in Chapter II, Conceptual Blending Theory has three operations: composition, completion, and elaboration. Composition attributes outer relations between mental spaces. Completion uses generic spaces, an existing knowledge base, and experience. Elaboration blends input mental space information with generic spaces, finds an emergent structure, and projects this structure to blend space [31].

One of these blending operations in the ADL Simulation is in detecting merge activity. Figure 40 shows the merge detector ticket execution.

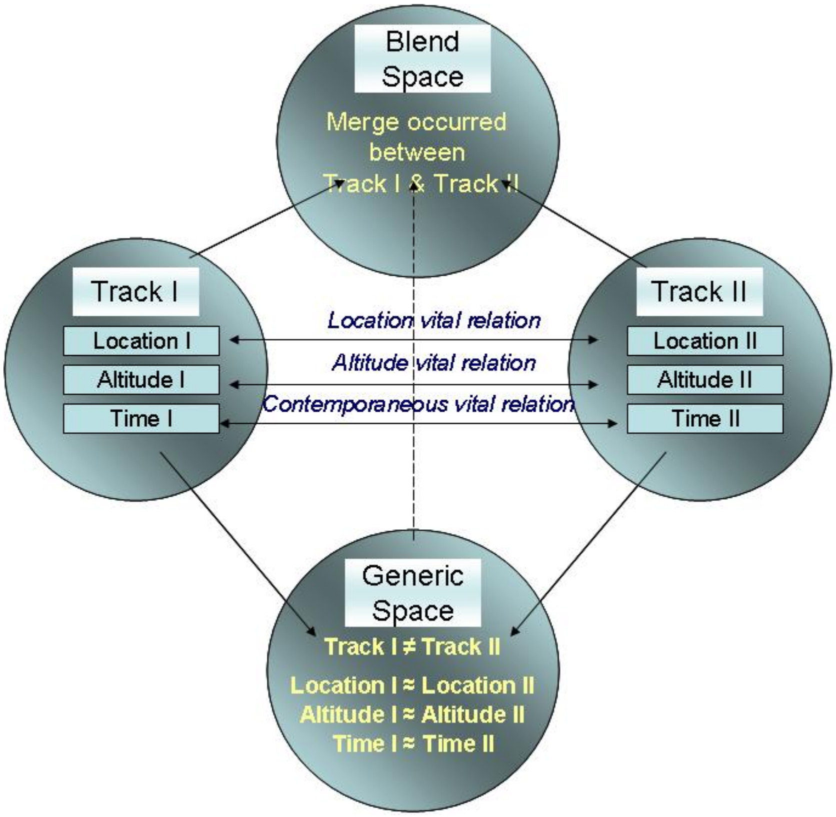


Figure 34. Merge Detector Blending Operation

In the blend operation shown above, the ADL Simulation defines three outer vital relations. They are space vital relations including location information, altitude information and the contemporaneous time vital relation. The input mental spaces are the two different tracks. Under the rules of generic space, merge activity is projected to blend space. The organizing frame for the two input spaces is an air track organizing frame. Only location, altitude and time elements of mental spaces participate in the composition operation of blending theory. Later these three elements perform the completion operation of blending theory by applying the values of elements with the rules of generic space. If rules match, merge activity is projected onto blend space. This blending operation exemplifies the mirror network of the four network types of Gilles and Turner since both organizing frames are same [32].

In the snooper supported coordinated attack scenario, we used the cause-effect vital relation in blending. The cause is the snooper and effect is the upcoming attack operation on ship. The emergent structure in blend space is a coordinated attack of at least two aircraft. The attacker aircraft should be in the environment after the snooper is observed. The threat level is broadcast to all predictor agents. This is called a generalization operation in blending theory. Making a decision to increase the threat level is another blending operation where part-whole vital relation is used. The whole is the coordinated attack on ship supported by the snooper. Since the regional agent recognizes the whole, it increases the threat level. This is the third operation of blending theory, elaboration.

We defined three of the blending operations of the ADL Simulation above. These blending operations are parallel to linguistic blending operations that we described in Chapter II. Besides their vital relations we created our own vital relations in the ADL Simulation. Composition is one of the operations of blending theory and finding these relations is the focal point of performing composition operation. Attributing these vital relations is another important point of operation. We managed to link these elements of input mental spaces via CMAS library in the simulation. Connectors and corresponding queries are the relations between different mental spaces. In case of a split operation, when an air track is created, Track II extends its location connector. This connector finds a match with corresponding query of all other tracks in the same membrane and blending operation is performed. When a split operation is found between two tracks the predictor agents then change their tolerance limit for changes on the behavior of the air track. This is called generalization in blending theory.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. RESEARCH QUESTIONS RESULTS AND THE EVAULATION OF THE SIMULATION

A. RESEARCH QUESTIONS

1. Overview

During the research we examined four sets of issues: the level of reality of the ADL Simulation, the level of precision of decisions given by the model, the level of closeness to the decisions given by the air-defense personnel, and the effect of analog versus digital decision-making processes on the simulator. But the model is not ready to be embedded into current tactical warfare systems because more careful work is needed on many of the details. Our purpose is to get insight into the decisionmaking process and to show the possibility of implementing a model working close to the way that the human brain works for a specific task.

1. General Testing Methodology

Tests used the following default variables:

Range threshold value:	25 nm
Speed threshold value:	500 knots
CPA threshold value:	15 nm

We ran the simulation 10 times for each test, which resulted in 190 runs. 10 runs tested the level of reality of the ADL Simulation, 90 runs tested the level of closeness to the way human brain works with analog decision-making, and 90 runs tested digital decisionmaking. We limited each scenario time period to 5-6 minutes. The scenarios were as in Table 13.

Scenario No	Scenario
1	5 civilian aircraft with/without threat intelligence
2	3 civilian aircraft and 1 friendly aircraft with/without threat intelligence
3	3 civilian, 1 friendly, and 1 hostile aircraft with/without threat intelligence
4	2 civilian, 1 hostile, 1 snooper, and 1 friendly aircraft with/without threat intelligence
5	3 civilian aircraft and a coordinated detachment attack with/without threat intelligence
6	3 civilian aircraft and a missile attack with/without threat intelligence
7	3 civilian aircraft and a terrorist attack with/without threat intelligence
8	3 civilian aircraft, 1 missile, and a coordinated detachment attack with/without threat intelligence
9	3 civilian aircraft and a terrorist attack with/without threat intelligence

Table 13. Scenarios Used in the Simulation Test and Analysis

We used an approximate uniform distribution of Java API for random number selection.

B. THE LEVEL OF REALITY OF THE ADL SIMULATION

We allocated time to implement a realistic user interface and environment for the ADL Simulation as much as we did for the implementation of the cognitive model. We believed that only a simulated environment as close as possible to a real environment would give us accurate results. In this test we analyzed how well the real track agents behave based on their roles in the simulation. The ADL Simulation was tested by two air-warfare officers (AAWO), two principal warfare officers (PWO), and 3 Air Force pilots. We ran the simulation ten times with different scenarios for each subject in tests. In general all of them supported the reality of the simulation. Their main criticisms were:

- One expert criticized the lack of issuing warnings to air contacts.
- Two experts stated that it would be more realistic if the ship had movement capability.
- Two experts criticized the lack of a task air-defense missions. We restricted the simulation to only one ship, but agree that air defense is not the responsibility of only one ship. Information transformation via tactical systems is paramount for establishing a real-time tactical air picture. However to simplify the simulation we eliminated Link services.
- Three experts criticized the reference system used in the simulation. We used an (x,y) coordinate system and avoided real-world reference systems to minimize the computation in the simulation.
- Four experts declared that the civilian, snooper, coordinated detachment attack, and missile-attack agents behaved as they should. They said that the behaviors of the friendly and hostile aircraft could be made more realistic. We agree with that criticism. However since this research is unclassified, we avoided real attack scenarios and missions but used the attack scenarios in game technologies.

The experts confirmed that the simulation is close to a real environment and its capabilities. However we had to simplify our simulation in some cases to decrease the computation.

C. THE ACCURACY OF THE DECISIONS BY THE ADL SIMULATION

We recorded the actual identities of the aircraft in each simulation and then compared them with the predictor track agent's predicted identities. We ran the simulation ten times for each nine different scenarios. Table 14, Table 15, and Table 16 show the times in seconds for the model to identify the contacts correctly.

Time to Identify Civilian	Analog process without Threat Intelligence	Analog Process with Threat Intelligence	Digital Process without Threat Intelligence	Digital Process with Threat Intelligence
Mean	17.33	18.38	17.98	20.479
Standard Deviation (s)	5.68	7.315	4.665	5.157
Variance (s^2)	32.269	53.519	21.77	26.597

Table 14. Civilian Aircraft ID Results of Tests

Table 14 shows the identification time of a civilian aircraft under four different circumstances. We found out that when a threat is expected, the time to identify a civilian aircraft is increased. When analog processing techniques were used, the model identified the civilian aircraft more quickly. With analog techniques we also found out that the weights of the competing models were close to each other. We believe that by using analog processing techniques, the system is more stable because the weights of competing models were kept close to each other and there is a smooth transition between competing models.

Time to Identify Friendly	Analog process without Threat Intelligence	Analog Process with Threat Intelligence	Digital Process without Threat Intelligence	Digital Process with Threat Intelligence
Mean	11.139	10.74	13.09	12.599
Standard Deviation (s)	5.324	4.135	4.722	4.606
Variance (s^2)	28.347	17.1	22.301	21.222

Table 15. Friendly Aircraft ID Results of Tests

Table 15 shows the time that the simulation took to identify the friendly aircraft under four different circumstances. The results show that when a threat is expected, both processing techniques identify the friendly aircraft in a shorter time. This result supports the results of Liebhaber's research. Another result is that an analog processing technique identifies the friendly aircraft in a shorter time. A further result is that the standard deviation is decreased when a threat is expected in the environment.

6 Minute Scenario 1 Hostile Track		**Analog process without Threat Intelligence**	**Analog Process with Threat Intelligence**	**Digital Process without Threat Intelligence**	**Digital Process with Threat Intelligence**
Time to Identify Hostile (sec)	**Mean**	11.3	11.52	13.3	11.059
	Standard Deviation (s)	0.82	1.541	1.232	0.482
	Variance (s²)	0.674	2.376	1.519	0.233
Total Time aircraft is identified as Hostile (sec)	**Mean**	180.66 (51.21%)	274.66 (76.3%)	195.959 (56%)	114.16 (31.7%)
	Standard Deviation (s)	17.093	89.879	61.245	57.047
	Variance (s²)	313.069	8078.37	3751.05	3254.452
Total Time aircraft is identified as Suspect (sec)	**Mean**	157.836 (44.74%)	63.44 (17.62%)	135.94 (38%)	227.16 (63.1%)
	Standard Deviation (s)	16.807	89.882	59.459	55.257
	Variance (s²)	282.501	8078.89	3535.427	3053.343

Table 16. Hostile Aircraft ID Results of Tests

Table 16 shows the time that the simulation requires to identify an aircraft as hostile and suspect, and the time that the simulation requires to identify an aircraft as hostile under four different circumstances. The results show that analog processing techniques identify the hostile activity in a shorter time. When a threat is expected the model identifies the hostile activities in a shorter time. During the tests, we found out that the simulation identifies hostile activities as hostile and suspect the majority of the time. The tests showed analog processing techniques identify hostile activity faster than the digital processing techniques do. When the threat is expected, the digital processing techniques identified the hostile activities less than suspect identification. We observed that the simulation identified hostile activities in close range, especially when the track is in the range threshold value which has a 25 nm default. Another interesting result achieved during the tests is that when the aircraft is far from the ship, the model tends to identify the air contact as suspect, but as the contact approaches the ship the model changes the identification from suspect to hostile.

D. THE LEVEL OF CLOSENESS OF DECISIONS GIVEN BY THE MODEL TO THE DECISIONS GIVEN BY THE EXPERTS

We worked with two PWOs and one AAWO while testing the closeness of the decisions of the model to decisions of the real air warfare personnel. We ran each of nine simulations for each of the expert twice, once with threat intelligence and once without threat intelligence. We asked them to talk continuously while they made decisions to catch the factors affecting the decisionmaking process. We recorded their voice on a tape recorder. During the tests, the datalog option was also kept “On”. We then compared the factors used by the user with the factors used by the model.

The results showed that all the factors used by the subjects were a subset of the factors defined in the ADL Simulation. However it was clear that ADL Simulation was ten times faster than human decisionmaking on the average. That proves our motivation for the ADL Simulation that we need computers with their high speed processing capabilities in time sensitive areas such as air-defense of a naval unit.

The results of the tests also showed that decisions made by the ADL Simulation were the same as the experts made. We also observed that the procedure of the experts in threat assessment is checking the factors affecting the decisionmaking and comparing them with expected values for each identity in their minds, as observed in Liebhaber’s research.

VII. FUTURE WORK AND DEVELOPMENT OF THE AIR DEFENSE LABORATORY SIMULATOR

A. FUTURE WORK INTRODUCTION

The ADL Simulation was inspired by a previous thesis written by Sharif Calfee. We believe that ADL Simulation will have a similar effect on the subsequent research. In fact our second goal of replacing the human factor in threat assessment could be accomplished in the next few years. The ADL Simulation has also reached the point in which we can create our integration network and traverse in the network in assisting the human air-defense officers with threat assessment.

B. DEVELOPMENT OF THE ENVIRONMENT

We simplified some of the details to reduce the computation time of the simulation. The simulation could be enhanced upon by adding movement capability to the ship. The environment could also be improved by adding a geographical reference system into the model. If this feature is added to the simulation, by which a user could also add actual maps to the simulation.

We defined only one surface ship in the environment. The simulation could be enhanced by adding more surface ships. The coordination between ships and the task-force air defense is another component to be examined in the development of the model. We defined the missions of friendly military aircraft as either a path or area mission, but did not specifically implement any of the missions. This may make the simulation more realistic in terms of the variety of aircraft behaviors.

C. INTEGRATED KNOWLEDGE TRANSFER

The ADL Simulation is able to create an integration network and retrieve the mental spaces by using the CMAS library. Each time the simulation is run, the integration network is created again in the simulation. A valuable improvement to the simulation is the transfer and addition of the created knowledge from another simulation's or another agent’s knowledge base. This can also be labeled as “experience transfer”. The problem of transferring a knowledge base to another is a compatibility issue. However the ADL Simulation uses the CMAS library to traverse the integration network. The only requirement of the second environment to be compatible with an attached knowledge base from another environment is the ability to use the CMAS library and use the same queries. The ADL Simulation is ready to transfer an isolated part of integration network to another agent in another environment. The transfer of an isolated part of the integration network created in one agent to another agent is a huge step for agents in gaining experience and then transferring this experience to other agents. This is like what teachers do to students at the school or what experienced personnel do to a new hire.

Transferring knowledge to another agent would enable us to explore another interesting research issue. It is clear that experienced air-defense personnel use a greater knowledge base than novice air-defense personnel use. The traditional method of seeing the effects of using a novice person in air-defense or any area will slow down the simulation process or extract certain numbers of rules from the simulation. However this process does not create the real results because slowing down the thinking process of an agent or banning an agent to use its existing knowledge base would not simulate the real world situations. In the future, if only a portion of the integration network is transferred to an agent, this portion would represent the novice air-defense personnel, we can get the realistic results from the simulation.

D. THE ADL SIMULATION AS A TRAINING TOOL

The ADL Simulation has two modes of operation: User mode or model mode. The user mode was originally implemented for test purposes. We used the user mode of the simulation to compare model-based decisions and human air-defense decisions. This could be improved to make the ADL Simulation a training tool for air-defense personnel. The model can determine the experience level of the air-defense personnel by measuring the level of usage of created integration network inside the simulation. While experienced air-defense personnel would use the entire integration network, novice personnel would use only a portion of the integration network of the model. The differences could be clue to determine the experience level of the air-defense personnel and lead us in a certain direction to train personnel.

E. IMPLEMENTING THE ADL SIMULATION WITH BAYESIAN METHODS

We used integration networks for the solution to the problem. There are other ways to model air defense. One of them is using a Bayesian method. The probability of an event in Bayesian method is the frequency of observed occurrence in a sample. A simulation could be developed by using Bayesian methods with probabilities obtained from actual air-defense exercises.

THIS PAGE INTENTIONALLY LEFT BLANK

VIII. SUMMARY AND CONCLUSION

The Air Defense Laboratory Simulation is a software program that models the way an air-defense officer thinks in the threat assessment process. It uses multi-agent system technology and is implemented in the Java programming language. We created integration network and modeled the decisionmaking process of an air-defense officer by using Conceptual Blending Theory and the CMAS library which implements it. The CMAS library has the facility of connectors and queries to create the integration network. Each node of the integration network is a mental space, information packets. These packets are connected to each other via connectors of CMAS library. The model of the Simulation can retrieve the required data of any mental space of the integration network and use them to create new mental spaces. Newly created mental space is then attached to end of integration network. We represented the integration network in a tree structure so that a human user can traverse on this tree and see the decisionmaking process step taken by the cognitive model.

The development of the ADL Simulation demonstrated that using computers in time-sensitive areas like air-defense as assistant to air-defense personnel improves the success rate. We demonstrated that the ADL Simulation is also faster than human decisionmakers and can be used as an assistant to them in threat assessment. In long term, the ADL Simulation might serve as a basis for replacement of humans in threat assessment.

We demonstrated that the usage of a blending theory originated in linguistic can be used in computer science field successfully. The usage blending theory in the ADL Simulation is not complete yet but we managed to start implementing our software with this theory. We are confident that by using blending theory our software can be implemented better.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

1. Gilles Fauconnier and Mark Turner, The Way We Think (New York: Basic Books, 2002).

2. Jacques Ferber, Multi-Agent Systems An Introduction to Distributed Artificial Intelligence, (Massachusetts: Addison-Wesley, 1999)

3. Fauconnier and Turner, 1

4. Seana Coulson and Gilles Fauconnier, “Fake Guns and State Lions: Conceptual Blending and Privative Adjectives.” in Cognition and Function in Language, ed. Fox, D. Jurafsky, and L. Michaelis (Palo Alto: University of Chicago Press, 1999), 1.

5. Fauconnier and Turner, 40.

6. Fauconnier and Turner, 40.

7. Fauconnier and Turner, 47.

8. Fauconnier and Turner, 48.

9. Seana Coulson and Cyma Van Patten, “Conceptual Integration and Metaphor: An Event-Related Potential Study,” Memory & Cognition (2002): 27.

10 Sally Wehmeier, Oxford Advanced Learner’s Dictionary (China: Oxford University Press, 2000), 171.

11. Fauconnier and Turner, 71.

12. Mark Turner and Gilles Fauconnier, “Conceptual Integration Networks”. Cognitive Science 22:1 (1998): 133-187.

13. Fauconnier and Turner, 92.

14. Fauconnier and Turner, 93.

15. Fauconnier and Turner, 120.

16. Fauconnier and Turner, 123.

17. Fauconnier and Turner, 63.

18. Fauconnier and Turner, 125.

19. Richard D. Amori, "An Adversarial Plan Recognition System For Multi-Agent Airborne Threats." In Symposium on Applied Computing (1992): 500

20. Michael J. Liebhaber and C.A.P. Smith, "Naval Air Defense Threat Assessment: Cognitive Factors Model," Command and Control Research and Technology Symposium (1999): 2

21. Michael J. Liebhaber and C.A.P. Smith, 14

22. Michael J. Liebhaber and Bela Feher, "Air Threat Assessment: Research, Model, and Display Guidelines", Pacific Science and Engineering Group (2002): 4

23. Liebhaber and Feher, 11

24. Sharif H. Calfee, *Autonomous Agent-Based Simulation of an Aegis Cruiser Combat Information Center Performing Battle Group Air-Defense Commander Operations* Thesis, Naval Postgraduate School, Monterey, California, March 2003, 119

25. David L. Hall and Hames Llinas, "An Introduction to Multisensor Data Fusion," in Proceedings of IEEE 85 (1997): 8

26. J. E. Hiles et. al., "StoryEngine: Dynamic Story Production Using Software Agents That Discover Plans," unpublished technical paper, Naval Postgraduate School, 2002.

27. B. A. Osborn, "An Agent-Based Architecture for Generating Interactive Stories," Ph.D. thesis, Department of Computer Science, Naval Postgraduate School, Monterey, California, September 2002.

28. Prima Games Strategy Guide, "Prima Fast Track Guides," <http://www.primagames.com/strategy/guide/90> (1 March 2004).

29. Liebhaber and Smith, 1

30. Liebhaber and Smith, 10

31. Fauconnier and Turner, 40.

32. Fauconnier and Turner, 120.

33. Fauconnier and Turner, 16.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Commander
   Space and Naval Warfare Systems Center, San Diego
   San Diego, California

4. Anadolu University Library and Documentation Department
   Eskisehir, Turkey

5. Arastirma Merkezi Komutanligi
   Pendik, Istanbul, Turkey

6. Bilkent University Library
   Bilkent, Ankara, Turkey

7. Bogazici University The Institute of Science and Engineering
   Computer Engineering
   Bebek, Istanbul, Turkey

8. Deniz Kuvvetleri Komutanligi
   Personel Daire Baskanligi
   Bakanliklar, Ankara, Turkey

9. Deniz Kuvvetleri Komutanligi Kutuphanesi
   Bakanliklar, Ankara, Turkey

10. Deniz Harp Okulu Kutuphanesi
    Tuzla, Istanbul, Turkey

11. Istanbul Technical University
    Electronic and Electric Faculty, Department of Computer Engineering
    Ayazaga, Istanbul, Turkey

12. Istanbul Technical University
    Software Development Center
    Maslak, Istanbul, Turkey

13. Middle East Technical University
Department of Computer Engineering
Ankara, Turkey

14. Military Operations Research Society (MORS)
Alexandria, VA